

**INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
GOIÁS**

**DESENVOLVIMENTO DE UM MANIPULADOR MECÂNICO
ARTICULADO COM QUATRO GRAUS DE LIBERDADE, ACIONADO
COM SISTEMA EMBARCADO E MONITORADO COM APLICATIVO
SUPERVISÓRIO**

FABRÍCIO FERREIRA MENDONÇA

DEZEMBRO

2015

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE
GOIÁS**

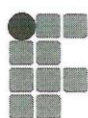
**DESENVOLVIMENTO DE UM MANIPULADOR MECÂNICO
ARTICULADO COM QUATRO GRAUS DE LIBERDADE, ACIONADO
COM SISTEMA EMBARCADO E MONITORADO COM APLICATIVO
SUPERVISÓRIO**

Trabalho de conclusão de curso apresentado à Banca Examinadora do Trabalho de Conclusão do Curso de Bacharel em Engenharia Elétrica, em cumprimento às exigências legais como requisito parcial à obtenção do título de Engenheiro Eletricista.

Prof. **Ghunter Paulo Viajante**, Dr. (IFG) – Orientador

Prof. **Jucélio Costa de Araújo**, Dr. (IFG)

Prof. **Wellington do Prado**, Msc. (IFG)



INSTITUTO FEDERAL
GOIÁS
Câmpus Itumbiara

MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE GOIÁS
CÂMPUS ITUMBIARA
DEPARTAMENTO DE ÁREAS ACADÊMICAS
COORDENAÇÃO DE CURSO DE BACHARELADO EM ENGENHARIA ELÉTRICA

TERMO DE APROVAÇÃO

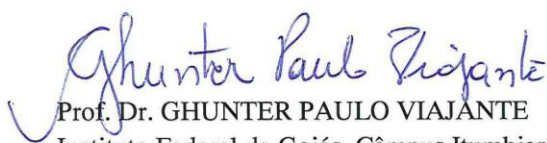
TÍTULO: DESENVOLVIMENTO DE UM MANIPULADOR MECÂNICO ARTICULADO COM QUATRO GRAUS DE LIBERDADE, ACIONADO COM SISTEMA EMBARCADO E MONITORADO COM APLICATIVO SUPERVISÓRIO

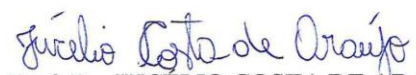
AUTOR: Fabrício Ferreira Mendonça

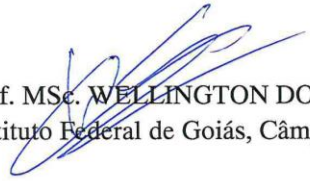
ORIENTADOR: Prof. Dr. Ghunter Paulo Viajante

Aprovado como parte das exigências para obtenção do diploma de Engenheiro Eletricista, do Instituto Federal de Goiás, Câmpus Itumbiara.

Itumbiara, 14 de dezembro de 2015.


Prof. Dr. GHUNTER PAULO VIAJANTE
Instituto Federal de Goiás, Câmpus Itumbiara


Prof. Dr. JUCÉLIO COSTA DE ARAÚJO
Instituto Federal de Goiás, Câmpus Itumbiara


Prof. MSc. WELLINGTON DO PRADO
Instituto Federal de Goiás, Câmpus Itumbiara

Ficha Catalográfica

M539d

Mendonça, Fabrício Ferreira

Desenvolvimento de um manipulador mecânico articulado com quatro graus de liberdade, acionado com sistema embarcado e monitorado com aplicativo supervisorio / Fabrício Ferreira Mendonça. -- Itumbiara, 2015.

134 f. : il. color. ; 30 cm.

Trabalho de conclusão de curso (Bacharelado em engenharia elétrica) - Instituto Federal de Educação, Ciência e Tecnologia de Goiás, *campus* Itumbiara, 2015.

Orientador: prof. Dr. Ghunter Paulo Viajante.

1. Robótica. 2. Sistemas de controle supervisorio. 3. Microcontroladores. 4. Cinemática das máquinas. I. Título. II. Viajante, Ghunter Paulo.

CDD 629.892

DEDICATÓRIA

A minha família e a todos os amigos que com muito carinho e apoio, não mediram esforços para que eu concluísse essa etapa da minha vida.

AGRADECIMENTOS

Este trabalho não teria sido possível sem o apoio e ao estímulo proporcionado por várias pessoas que ao longo do projeto contribuíram com ensinamentos, materiais e mão de obra durante a elaboração do experimento. Gostaria de fato expressar aqui meus profundos agradecimentos, pois ao longo do desenvolvimento houve tarefas difíceis em horários conturbados nos quais foram superados com a ajuda de todos.

Em primeiro lugar, agradeço a **Deus** pela força de vontade a mim atribuída, pela paciência, pelos conhecimentos e pela oportunidade de superar os momentos difíceis me fazendo tornar uma pessoa ainda mais madura e um profissional mais qualificado.

Ao meu professor orientador **Dr. Ghunter Paulo Viajante** pela orientação, paciência e dedicação durante todo o tempo dedicado a elaboração do projeto como um grande professor, amigo e mentor.

Ao **Instituto Federal de Goiás** pela oferta do curso de Engenharia Elétrica e pela disponibilização do espaço físico no qual todo o trabalho foi desenvolvido.

Ao meu pai **Nilson Mendonça da Silva** e à minha mãe **Sirley Ferreira Mendonça da Silva**, pela sólida formação e disciplina dada durante o meu crescimento que me proporcionou a continuidade nos estudos até à chegada deste final de curso, os meus eternos agradecimentos.

À minha noiva **Kely Deyse Ferreira da Silva** pelo apoio emocional e incentivo durante toda a minha jornada no curso, nos piores e melhores momentos do projeto, que esteve comigo e presenciou todo o progresso ocorrido durante a evolução do trabalho.

Ao meu grande companheiro de classe e eterno amigo **Ricardo Tirone Fidelis** por ter me auxiliado em todos os momentos de dificuldade acadêmica, pessoal e profissional, fica aqui minha profunda gratidão.

Aos colegas de trabalho **Maycon Ferreira** e **Neilton Leal**, que foram de muito valor durante a elaboração do manipulador que exigiu ajuda nos momentos de corte e montagem mecânica. Também agradeço ao amigo e companheiro de trabalho **Pablo Soares** pela flexibilidade no horário de trabalho me proporcionando a continuidade do curso e o desenvolvimento deste trabalho.

Aos colegas de graduação, muito obrigado pela amizade, pelo incentivo e principalmente pelo companheirismo.

Aos professores do IFG que, notavelmente, influenciaram nas tomadas de decisão e contribuíram muito durante o meu crescimento acadêmico. Em especial agradeço aos professores

e amigos **Marcos Antônio Arantes**, **Eric Nery Chaves**, **Ghunter Paulo Viajante** e **Cláudio Roberto Pacheco**, e as gloriosas professoras **Tatiana Pires Fleury Bezerra** e **Priscila Branquinho Xavier**.

RESUMO

O trabalho discorre sobre o desenvolvimento de um braço robótico fazendo uso de materiais de baixo custo possibilitado a elaboração de um projeto que envolve a área de desenvolvimento tecnológico.

O projeto trata do desenvolvimento de um braço robótico articulado, reutilizando alguns materiais sucateados, para separar peças de metal de peças plásticas que são transportadas através de uma esteira rolante. O manipulador robótico é controlado através de um sistema embarcado que envia informações para uma tela de supervisório onde é informado o status do processo, a quantidade de peças separadas por tipo e ainda é possível uma visualização em 3D dos movimentos realizados pelo braço robótico no campo físico.

Palavras-chave: Manipulador robótico, supervisório, microcontrolador, baixo custo.

LISTA DE FIGURAS

FIGURA 1.1 - ROBÔ PALETIZADOR [27].	15
FIGURA 1.2 - ESTIMATIVO DE CONSUMO DE UNIDADES ROBÓTICAS [1].	16
FIGURA 2.1 - ASPECTOS CONSTRUTIVOS DE UMA MÁQUINA CC [2].	20
FIGURA 2.2 - SERVOMOTOR [5].	24
FIGURA 2.3 - SENSORES ÓPTICOS [7].	26
FIGURA 2.4 - SENSORES DE PROXIMIDADE [8].	27
FIGURA 2.5 - PIC 16F877A.	31
FIGURA 2.6 - DIVERSIDADE ARDUINOS [11].	32
FIGURA 2.7 - CONFIGURAÇÃO PONTE H.	34
FIGURA 2.8 - CIRCUITO PARA ACIONAMENTO DA PONTE H.	35
FIGURA 2.9 - METODOLOGIA DE CONCEPÇÃO DE SISTEMAS MECÂNICOS PASSIVOS [6].	37
FIGURA 2.10 - ESTRUTURA DE UM SISTEMA MECATRÔNICO [6].	39
FIGURA 2.11 - JUNTA E VÍNCULOS.	40
FIGURA 2.12 - VÍNCULOS.	41
FIGURA 2.13 - GARRA DE DOIS DEDOS.	42
FIGURA 2.14 - ROBÔ TIPO CARTESIANO [6].	45
FIGURA 2.15 - ROBÔ CILÍNDRICO [6].	46
FIGURA 2.16 - ROBÔ POLAR [6].	46
FIGURA 2.17 - ROBÔ ARTICULADO [6].	47
FIGURA 2.18 - FLUXO DE COMUNICAÇÃO EM SUPERVISÓRIOS [12].	49
FIGURA 2.19 - EXEMPLO DE TELA DE SUPERVISÓRIO [13].	49
FIGURA 2.20 - EXEMPLO TELA FRONTAL LABVIEW [15].	51
FIGURA 2.21 - DIAGRAMA DE BLOCO E PAINEL FRONTAL [15].	51
FIGURA 3.1 - VISUALIZAÇÃO DO PROJETO.	53
FIGURA 3.2 - SUPERVISÓRIO DESENVOLVIDO.	53
FIGURA 3.3 - DIAGRAMA ESQUEMÁTICO DO PROJETO.	54
FIGURA 3.4 - MOTOR DE VIDRO ELÉTRICO UTILIZADO NO OMBRO.	56
FIGURA 3.5 - MOTOR LIMPADOR DE PARA-BRISA UTILIZADO NA BASE.	57
FIGURA 3.6 - MOTOR DC UTILIZADO NO COTOVELO.	57
FIGURA 3.7 - MOTOR DC UTILIZADO NO PUNHO.	57
FIGURA 3.8 - MODULAÇÃO DE CONTROLE PWM PARA SERVOMOTORES DE BAIXA POTÊNCIA [16].	58
FIGURA 3.9 - SERVOMOTOR UTILIZADO NA GARRA [16].	58
FIGURA 3.10 - PRIMEIRA OPÇÃO DE LINK DE SUSTENTAÇÃO.	59
FIGURA 3.11 - ESTRUTURA FINAL DEFINIDA PARA SEGUIR COM O PROJETO.	60
FIGURA 3.12 - CONJUNTO DA BASE.	61
FIGURA 3.13 - ARTICULAÇÃO OMBRO.	61

FIGURA 3.14 - ARTICULAÇÃO COTOVELO.....	62
FIGURA 3.15 - ARTICULAÇÃO PUNHO.....	62
FIGURA 3.16 - ACABAMENTO FINAL.....	63
FIGURA 3.17 - DESENHO DO MANIPULADOR EM PLATAFORMA AUTOCAD.....	64
FIGURA 3.18 - DRIVE ACIONAMENTO BASE / OMBRO.....	66
FIGURA 3.19 - ACIONAMENTO SERVOMOTOR.....	66
FIGURA 3.20 - CIRCUITO DIVISOR DE TENSÃO.....	67
FIGURA 3.21 - ADAPTAÇÃO DOS SENSORES PARA A BASE.....	68
FIGURA 3.22 - VISTA EM CORTE DA POSIÇÃO DOS SENSORES.....	68
FIGURA 3.23 - CIRCUITO TIL32 E TIL78.....	69
FIGURA 3.24 - ACIONAMENTO DOS DISPOSITIVOS EXTERNOS.....	69
FIGURA 3.25 - CIRCUITO PULL UP E ANALÓGICAS.....	70
FIGURA 3.26 - SINAL FORNECIDO PELO POTENCIÔMETRO COM TENSÃO DE REFERENCIA FORNECIDA PELA FONTE.....	71
FIGURA 3.27 - SINAL FORNECIDO PELO POTENCIÔMETRO COM TENSÃO DE REFERÊNCIA FORNECIDO PELO ARDUINO.....	71
FIGURA 3.28 - ESQUEMA DA PONTE H NO ISIS PROTEUS.....	72
FIGURA 3.29 - LAYOUT DA PLACA DISPONIBILIZADA PELO ARES PROTEUS.....	73
FIGURA 3.30 - PROJETO PARA CONFEÇÃO DA PLACA.....	74
FIGURA 3.31 - FRENTE E VERSO DA PLACA ELETRÔNICA PRONTA.....	74
FIGURA 3.32 - CIRCUITO ELETROELETRÔNICO.....	75
FIGURA 3.33 - FLUXOGRAMA DA ROTINA PROGRAMADA.....	77
FIGURA 3.34 - MALHA DE CONTROLE DE POSIÇÃO DOS MOTORES.....	78
FIGURA 3.35 - SINAL PWM COM 78% DA VELOCIDADE.....	79
FIGURA 3.36 - SINAL PWM A 1KHz.....	80
FIGURA 3.37 - ENCAPSULAMENTO DE DADOS PARA COMUNICAÇÃO SERIAL.....	81
FIGURA 3.38 - PROTOCOLO DE COMUNICAÇÃO.....	83
FIGURA 3.39 - ESQUEMA DE COMUNICAÇÃO ENTRE ARDUINO E LABVIEW.....	83
FIGURA 3.40 - TRECHO RESPONSÁVEL POR INFORMAR AS POSIÇÕES DAS JUNTAS.....	84
FIGURA 3.41 - DESENHO 3D QUE REPRESENTA O BRAÇO FÍSICO [20].....	84
FIGURA 3.42 - DESCRIÇÃO DOS ITENS DA TELA DE SUPERVISIONAMENTO.....	85
FIGURA 4.1 - ESTEIRA DE ALIMENTAÇÃO DE PEÇAS.....	86
FIGURA 4.2 - SUPERVISÓRIO EM ESTÁGIO DE REFERENCIAMENTO.....	87
FIGURA 4.3 - SISTEMA HABILITADO AGUARDANDO PEÇAS.....	89
FIGURA 4.4 - SUPERVISÓRIO MOSTRANDO HISTÓRICO DE PEÇAS.....	89
FIGURA 4.5 - BRAÇO ROBÓTICO EM MOVIMENTO.....	90
FIGURA 4.6 - ASPECTO CONSTRUTIVO POTENCIÔMETRO.....	91
FIGURA 4.7 - SETPOINT X SAÍDA - OMBRO.....	91
FIGURA 4.8 - SETPOINT X SAÍDA - COTOVELO.....	92
FIGURA 4.9 - REDUÇÃO DA LARGURA DE PULSO.....	92
FIGURA 4.10 - SETPOINT X SAÍDA - PUNHO.....	93
FIGURA 4.11 - MEDIDAS VÍNCULOS.....	94

FIGURA 4.12 - CAPACIDADE ROTATIVA CINTURA.	94
FIGURA 4.13 – ABERTURA OMBRO EM 90°	95
FIGURA 4.14 - LIBERDADE COTOVELO.....	95
FIGURA 4.15 - LIBERDADE PUNHO.....	96

LISTA DE TABELAS

TABELA 2.1 - MODELOS DE MERCADO [28].....	32
TABELA 3.1 - TABELA E SEUS CORRESPONDENTES NUMÉRICOS.	82
TABELA 4.1 - VALOR GASTO NO PROJETO.	98
TABELA 4.2 - PREVISÃO DE GASTO COM MOTOR DE PASSO.	99

SUMÁRIO

CAPÍTULO 1	15
INTRODUÇÃO	15
1.1 OBJETIVOS	17
1.1.1 <i>Objetivo Geral</i>	17
1.1.2 <i>Objetivos Específicos</i>	17
1.1.3 <i>Estrutura do Trabalho</i>	18
CAPÍTULO 2	19
FUNDAMENTAÇÃO TEÓRICA	19
2.1 MÁQUINAS ELÉTRICAS ROTATIVAS	19
2.1.1 <i>Máquinas de Corrente Contínua</i>	20
2.1.2 <i>Máquinas de Corrente Alternada</i>	22
2.1.3 <i>Servomotores</i>	23
2.2 ELEMENTOS ELÉTRICOS DE COMANDO	24
2.3 ELEMENTOS SENSORIAIS	25
2.4 SISTEMAS EMBARCADOS	28
2.4.1 <i>Microprocessadores</i>	28
2.5 CIRCUITO PONTE H	33
2.6 SISTEMAS MECATRÔNICOS	35
2.6.1 <i>Sistemas Mecânicos Passivos</i>	35
2.6.2 <i>Sistemas Mecânicos Ativos</i>	37
2.6.3 <i>Sistemas Ativos com Realimentação Elétrica</i>	38
2.7 ASPECTOS CONSTRUTIVOS DE MANIPULADORES	39
2.7.1 <i>Juntas Robóticas</i>	39
2.7.2 <i>Vínculos</i>	41
2.7.3 <i>Garras e Ferramentas</i>	41
2.7.4 <i>Precisão e Repetição</i>	43
2.7.5 <i>Acoplamentos Mecânicos</i>	43
2.7.6 <i>Formas de Acionamento dos Manipuladores</i>	43
2.8 CLASSIFICAÇÃO DOS ROBOS	44
2.8.1 <i>Coordenadas Cartesianas</i>	45
2.8.2 <i>Coordenadas Cilíndricas</i>	45
2.8.3 <i>Coordenadas Polares (Esféricos)</i>	46
2.8.4 <i>Coordenadas de Revolução (Articulado)</i>	47

2.9 SISTEMAS SUPERVISÓRIOS	47
2.9.1 Sistema SCADA (<i>Supervisory Control and Data Acquisition</i>).....	48
2.9.2 LabVIEW	50
CAPÍTULO 3	52
DESENVOLVIMENTO DO PROJETO.....	52
3.1 ETAPAS PARA CONSTRUÇÃO DO BRAÇO ROBÓTICO	55
3.1.1 Escolha do Motor.....	55
3.1.2 Estrutura Mecânica do Manipulador – Vínculos.....	58
3.1.3 Construção e Montagem do Manipulador Robótico	60
3.2 ESTRUTURA ELETROELETRÔNICA.....	64
3.2.1 Drivers de Acionamento dos Motores – Ponte H	65
3.2.2 Circuito Elétrico do projeto	67
3.2.3 Placa Eletrônica de Acionamento do Manipulador	71
3.2.4 Programação e Controle Braço Robótico.....	75
3.3 PLATAFORMA DE SUPERVISÃO	81
3.3.1 Protocolo de Comunicação	81
3.3.2 Tela supervisão	83
CAPÍTULO 4	86
RESULTADOS EXPERIMENTAIS	86
4.1 REFERENCIAMENTO ESPACIAL DO MANIPULADOR	87
4.2 OPERAÇÃO E SUPERVISÃO	88
4.3 CONTROLE DE POSIÇÃO DOS MOTORES	90
4.4 ALCANCE DE ATUAÇÃO DO BRAÇO	93
4.5 LEVANTAMENTO DE CUSTOS.....	96
CAPÍTULO 5	100
CONSIDERAÇÕES FINAIS.....	100
5.1 SUGESTÃO DE TRABALHOS FUTUROS	101
APENDICE A – CÓDIGO FONTE.....	102
REFERÊNCIAS BIBLIOGRÁFICAS.....	132

CAPÍTULO 1

INTRODUÇÃO

Ao longo dos últimos anos o homem vem buscando aprimorar os processos produtivos fazendo uso da tecnologia se baseando nos fundamentos robóticos de manipulação por meio da Automação Industrial. Uma das técnicas é utilizar manipuladores mecânicos para executar tarefas industriais repetitivas, precisas e que exigem um alto aspecto dinâmico, envolvendo qualidade e velocidade de produção.

Esse desenvolvimento vem sendo possível graças aos avanços da microeletrônica, em conjunto com a mecânica, que desenvolvem projetos com velocidade e robustez para suprir as necessidades industriais. Podem ser controlados por comandos numéricos, como as máquinas CNC, por controladores lógicos programáveis, neste caso os CLP's e visualizados e projetados através dos softwares de informática como CAD e o CAM.

O processo de utilização de manipuladores nas indústrias ganhou muita força nos últimos 40 anos e estão espalhados por todo o mundo com uma infinidade de aplicações. No Brasil já existem empresas que trabalham no ramo da robótica industrial desenvolvendo manipuladores para o mercado. A figura 1.1 destaca um braço robótico destinado a paletização automática fornecida pela Torfresma, uma empresa que propaga tecnologia brasileira situada no estado de Santa Catarina.



Figura 1.1 - Robô paletizador [27].

A robótica contempla aplicações industriais, domésticas, trabalhos de riscos, inspeções submarinas e aplicações no ramo da medicina.

São caracterizados como manipuladores aqueles que possuem forma de um braço, contendo geralmente quatro ou mais graus de liberdade. Possuem articulações, braços, punhos e mãos para segurar os objetos manipulados. As indústrias automobilísticas fazem uso da metade dos robôs manipuladores vendidos e espalhados pelo mundo e são as principais fontes de pesquisa e desenvolvimento neste seguimento [1]. A figura 1.2 mostra uma estimativa de crescimento da robótica nas indústrias separadas por aplicação.

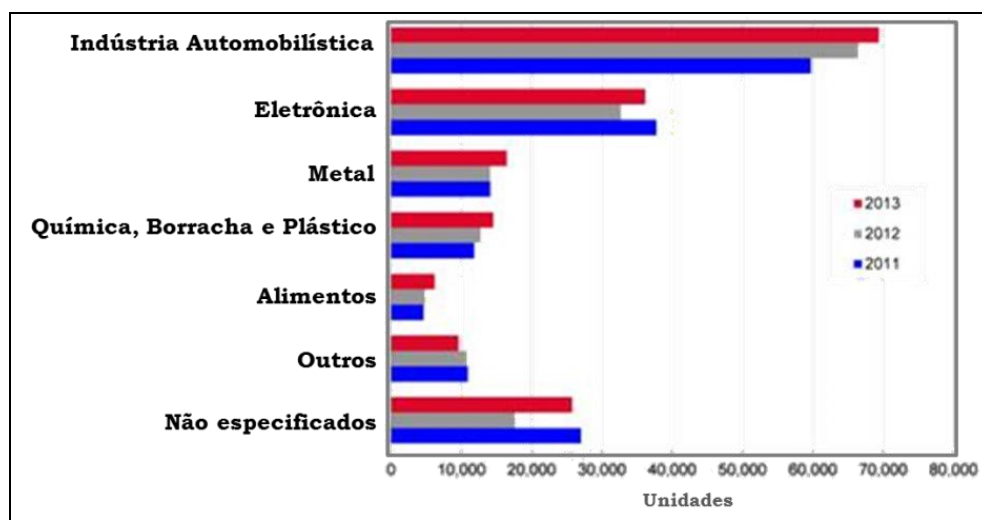


Figura 1.2 - Estimativo de consumo de unidades robóticas [1].

Abordando o contexto da automação industrial e da mecatrônica, este trabalho consiste no desenvolvimento de um manipulador com a sistemática parecida com os manipuladores robóticos industriais, sendo capaz de simular em sala de aula um processo industrial semelhante aos incorporados nas indústrias. A separação das peças e o supervisionamento podem ser caracterizados como próximos da realidade empregada nas empresas que fazem uso desta tecnologia.

O projeto é interdisciplinar, pois envolve estudos de diversas áreas como: circuitos elétricos, máquinas elétricas, resistência dos materiais, eletrônica analógica, eletrônica digital, sistemas de controle, robótica, sistemas microcontrolados, redes industriais e programação de computadores.

1.1 OBJETIVOS

O trabalho consiste na implementação de um braço mecânico controlado via sistema embarcado que consiga simular o processo industrial com o máximo de realidade possível para que sirva como base de projetos futuros e se torne um instrumento de ensino aplicado da robótica nos cursos de engenharia do IFG câmpus Itumbiara.

1.1.1 Objetivo Geral

- Desenvolvimento de um manipulador mecânico articulado com quatro graus de liberdade;
- Desenvolvimento um sistema embarcado para acionamento e controle do manipulador;
- Implementação de um supervisório para monitoramento e acionamento do manipulador;
- Utilização de dispositivos eletrônicos para confecção da placa de controle e acionamento.

1.1.2 Objetivos Específicos

- Estudar o funcionamento da máquina de corrente contínua e sua aplicação no cotidiano;
 - Estudar as formas de acionamento dessas máquinas;
 - Estudar os sensores disponíveis para utilização em automação e robótica e seus respectivos princípios de funcionamento;
 - Obter informações sobre qual o melhor dispositivo para controlar o projeto estudando os sistemas embarcados atuais;
 - Verificar a sistemática de sistemas mecatrônicos em relação aos seus aspectos construtivos e uma forma barata de construí-lo;
 - Pesquisar, estudar e desenvolver um aplicativo para trabalhar como supervisório de monitoração do projeto;
 - Viabilizar o custo para desenvolvimento do projeto;
 - Apresentar resultados e propor melhorias para projetos futuros.
-

1.1.3 Estrutura do Trabalho

Para alcançar os objetivos expostos, este trabalho foi organizado da seguinte forma:

O **capítulo 2** mostra as referências teóricas utilizadas para definição do melhor método construtivo considerando a viabilidade de custo do projeto. Apresenta o princípio de funcionamento dos principais dispositivos utilizados e os meios que são empregados na prática.

O **capítulo 3** mostra detalhadamente o projeto e o desenvolvimento do manipulador mecânico. Apresenta o circuito elétrico e eletrônico, o microcontrolador utilizado, descreve a rotina de programação do sistema embarcado e ainda o desenvolvimento do aplicativo supervisor.

No **capítulo 4** é possível visualizar os resultados experimentais obtidos com o projeto observando a forma de referenciamento espacial, o controle de posição dos motores, a operação da plataforma do experimento, a comunicação do dispositivo controlador com o computador, a área de trabalho do manipulador, o funcionamento do aplicativo supervisor e um levantamento de custo dos materiais utilizados.

CAPÍTULO 2

FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta os aspectos teóricos dos materiais, dispositivos, equipamentos e softwares utilizados para desenvolver o projeto.

2.1 MÁQUINAS ELÉTRICAS ROTATIVAS

As máquinas elétricas rotativas são equipamentos que convertem energia elétrica em energia mecânica ou vice-versa. Os motores são utilizados como fonte de conversão mecânica em elétrica e os geradores realizam a tarefa oposta. As máquinas são construídas de forma tal a realizarem com a máxima eficiência possível este processo de conversão. Elas possuem basicamente duas partes, onde uma delas é fixa, chamada de estator e uma parte móvel montada sobre um eixo situada no interior da máquina chamada de rotor, permitindo que exista rotação devido a sua fixação ser apoiada em rolamentos rotacionais.

Para o gerador, a rotação é suprida por uma máquina primária (uma fonte de energia mecânica) para produzir o movimento relativo entre os enrolamentos e o campo magnético da máquina fazendo com que a variação do fluxo magnético induzido nos enrolamentos permita a circulação de elétrons e trabalhe fornecendo energia elétrica para a carga. Para o motor, a energia elétrica é fornecida aos condutores da máquina gerando campo magnético e convertendo-o em energia mecânica rotacional.

As máquinas elétricas rotativas assumem diversas formas e são conhecidas por diversos nomes: Corrente Contínua, Síncronas, de Imã Permanente, de Indução, de Relutância Variável, de Histerese, sem escovas, e assim por diante [2].

2.1.1 Máquinas de Corrente Contínua

As máquinas de corrente contínua são caracterizadas pela sua versatilidade por meio das diversas combinações de enrolamentos de campo, excitados em derivação, série ou independentemente. Elas podem ser projetadas de modo a apresentar uma ampla variedade de características de tensão versus corrente ou de velocidade versus conjugado, para operações dinâmicas e em regime permanente.

Devido à facilidade com que podem ser controladas, as máquinas de corrente contínua têm sido usados frequentemente em aplicações que exigem uma ampla faixa de velocidades ou de controle preciso de posição. Nos últimos anos a eletrônica de potência vem sendo usada e trabalhada nos acionamentos de máquinas de corrente alternada (CA), e com isso, a tecnologia se desenvolveu de forma suficiente para substituir as máquinas CC por máquinas CA.

As máquinas CC tradicionais são compostas por Rotor, Estator, Escovas, Sapatas polares, Bobinas de Armadura (situadas no rotor) e Bobinas de Campo (situadas no estator). O estator tem pólos salientes e é excitado por uma ou mais bobinas de campo. O rotor da armadura das máquinas de CC tem quatro funções principais: Permitir rotação para ação geradora ou ação motora mecânica, produzir a ação de chaveamento necessária para a comutação, conter os condutores que induzem a tensão ou providenciar um torque eletromagnético e estabelecer uma faixa de baixa relutância para o fluxo [3]. A figura 2.1 mostra a composição mecânica de uma máquina de corrente contínua.

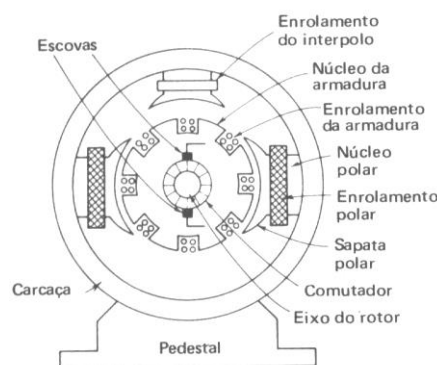


Figura 2.1 - Aspectos construtivos de uma máquina CC [2].

O enrolamento de campo pode conter poucas espiras de fio fino ou muitas espiras de fio grosso dependendo se a ligação será feita de forma serial ou na configuração de campo-shunt.

Estes enrolamentos são suportados pelos pólos nas sapatas polares que em sua forma curvada, permite espalhar o fluxo de forma mais uniforme dentro da máquina. No geral, para uma máquina de corrente contínua tradicional é necessário dois circuitos independentes para circulação de corrente e formação de campo girante. Para isso, deve-se existir um controle do circuito de campo e um controle para o circuito da armadura situada no rotor da máquina [3].

Existem também as máquinas de corrente contínua de ímã permanente que são encontradas em uma grande variedade de aplicações de baixa potencia. O enrolamento de campo é substituído por um ímã permanente, resultando em uma construção mais simples. Para essas aplicações, os ímãs permanentes oferecem uma série de benefícios úteis. A principal é que os ímãs não necessitam de excitação externa nem dissipam a potência correspondente para criar campos magnéticos na máquina. O espaço necessário para os ímãs permanentes pode ser inferior ao exigido pelos enrolamentos de campo e, assim, as máquinas de ímã permanente podem ser menores e, em alguns casos, de custo inferior ao de seus similares com excitação externa.

O rotor desse motor consiste em uma armadura CC convencional com lâminas do comutador e as escovas. Em uma das extremidades, há também um pequeno ímã permanente que produz o campo para um tacômetro CA. Esse pode ser usado em aplicações em que é necessário controle de velocidade. Diferentemente de uma máquina CC, cuja excitação de campo é externa com uma estrutura que se caracteriza pela presença de pólos salientes, os motores de ímã permanente geralmente têm uma estrutura de estator lisa consistindo em uma carcaça cilíndrica externa de material magnético permanente magnetizado com sentido radial. O rotor tem comutador, escovas e ranhuras para os enrolamentos, como em todas as máquinas CC. A carcaça externa serve a um duplo propósito, é feito de material magnético servindo assim de caminho de retorno para o fluxo magnético e de suporte para os ímãs [2].

O controle de velocidade das máquinas de corrente contínua baseia-se na circulação de corrente na armadura, que por sua vez, é determinada pela tensão aplicada em seus terminais. Para máquinas CC com enrolamentos de campo é necessário aplicar uma fonte de corrente contínua para criar o campo magnético e outra fonte de tensão variável para injetar corrente no circuito do rotor chamado de armadura. Em máquinas de ímãs permanentes é necessário apenas aplicar tensão nos terminais de armadura, pois o circuito de campo já está produzindo fluxo devido a fixação dos ímãs facilitando as técnicas de controle e de acionamento.

2.1.2 Máquinas de Corrente Alternada

Existem basicamente dois tipos de máquinas caracterizadas como máquinas CA, são elas as máquinas síncronas e as máquinas de indução. Nas máquinas síncronas, as correntes do enrolamento do rotor são fornecidas através de contatos rotativos fixados diretamente na parte estacionária do motor. Nas máquinas de indução, as correntes são induzidas nos enrolamentos do rotor por meio da combinação da variação, no tempo, de correntes no estator e do movimento do rotor em relação ao estator.

Com raras exceções, o enrolamento de armadura de uma máquina síncrona localiza-se no estator, e o enrolamento de campo, no rotor, esse também é o caso da máquina simplificada. O enrolamento de campo é excitado por uma corrente contínua que é levada até ele por meio de escovas estacionárias de carvão que fazem contato com anéis coletores ou anéis deslizantes girantes. É vantajoso ter o enrolamento de campo, único e de baixa potência, no rotor e o enrolamento de armadura, de potência elevada e geralmente trifásica, no estator.

A medida que o rotor gira, o fluxo concatenado do enrolamento da armadura varia no tempo. Tendo em vista as suposições de distribuição senoidal da densidade de fluxo e de velocidade constante do rotor, a tensão resultante na bobina será senoidal no tempo. A tensão da bobina passa por um ciclo completo a cada revolução. A sua frequência em ciclos por segundo (Hz) é a mesma que a velocidade do rotor em rotações por segundo: a frequência elétrica da tensão gerada está sincronizada com a velocidade mecânica, sendo essa a razão para a expressão máquina síncrona. Assim, uma máquina síncrona de dois pólos deve girar a 3600 rotações por minuto para produzir uma tensão de 60 Hz.

Quando uma máquina tem mais de dois pólos, é conveniente concentrar-se em apenas um par de pólos e assegurar-se de que as condições elétrica, magnéticas e mecânicas associadas aos demais pares de pólos sejam repetições das do par considerado. Por essa razão, é conveniente expressar os ângulos em graus elétricos ou radianos elétricos em vez de unidades mecânicas. Um par de pólos em uma máquina de múltiplos pólos, ou um ciclo de distribuição de fluxo, é igual a 360 graus elétricos ou 2π radianos elétricos [2].

No motor de indução a corrente alternada é fornecida diretamente ao estator, como em um transformador. Quando a excitação é feita por uma fonte polifásica equilibrada, um campo magnético é produzido no entreferro girando na velocidade síncrona. Essa velocidade é determinada pelo número de pólos do estator e pela frequência aplicada nos terminais da máquina. O motor de indução trifásico, chamado de gaiola de esquilo, possui o enrolamento que

consiste apenas em barras condutoras encaixadas em ranhuras no ferro do rotor e curto-circuitadas em cada lado por anéis condutores. A extrema simplicidade e a robustez da construção em gaiola de esquilo representam vantagens notáveis para esse tipo de motor de indução e, de longe, fazem dele o tipo de motor mais comumente usado. As máquinas de indução de rotor bobinado são relativamente incomuns, sendo encontradas apenas em um número limitado de aplicações especializadas [2].

Para controlar a velocidade uma máquina trifásica de rotor bobinado é necessário incluir bancos de resistência em série com o enrolamento do rotor para que a resistência da armadura aumente e conseqüentemente o escorregamento da máquina aumente também, tornando o controle de velocidade desse tipo de máquina muito robusto, complicado e aumenta a quantidade dos dispositivos de partida. Se tratando de um motor com rotor gaiola de esquilo a alteração da velocidade só pode ser feita de maneira eletrônica através de inversores de frequência que altera a frequência injetada nos terminais do motor alterando a velocidade de rotação do campo girante. Esse método é o mais adequado devido a resistência do rotor ser fixa e o número de pólos não poder ser modificado.

2.1.3 Servomotores

O Servomotor é um dispositivo eletromecânico possui uma parte fixa, também chamada de estator, e outra parte móvel conhecida como rotor, como muitas outras máquinas elétricas. O estator possui bastante semelhança ao de uma máquina elétrica convencional, porém com restrições quanto à alimentação, como será verificado posteriormente. O rotor é composto por ímãs permanentes, os quais são posicionados alinhadamente sobre o rotor e com o controlador, ou gerador de sinais, chamado de resolver. Devido a utilização dos ímãs permanentes, os servomotores podem proporcionar controle de velocidade, de posição e o torque no eixo, de forma constante e em larga faixa de rotação.

Para possibilitar tal controle é necessário o emprego do servoconversor, os quais são desenvolvidos especificamente para otimização dos Servomotores. Estes têm a função de controlar o fluxo eletromagnético e fornecê-lo para o Servomotor, pois os fluxos oriundos da rede não apresentam uma conformidade adequada para o acionamento dos servos, explicando também o bobinamento de estator diferenciado. Outro ponto importante a ser observado é que

cada servomotor possui o seu próprio servoconversor [4]. A figura 2.2 mostra as características internas de um servomotor industrial.

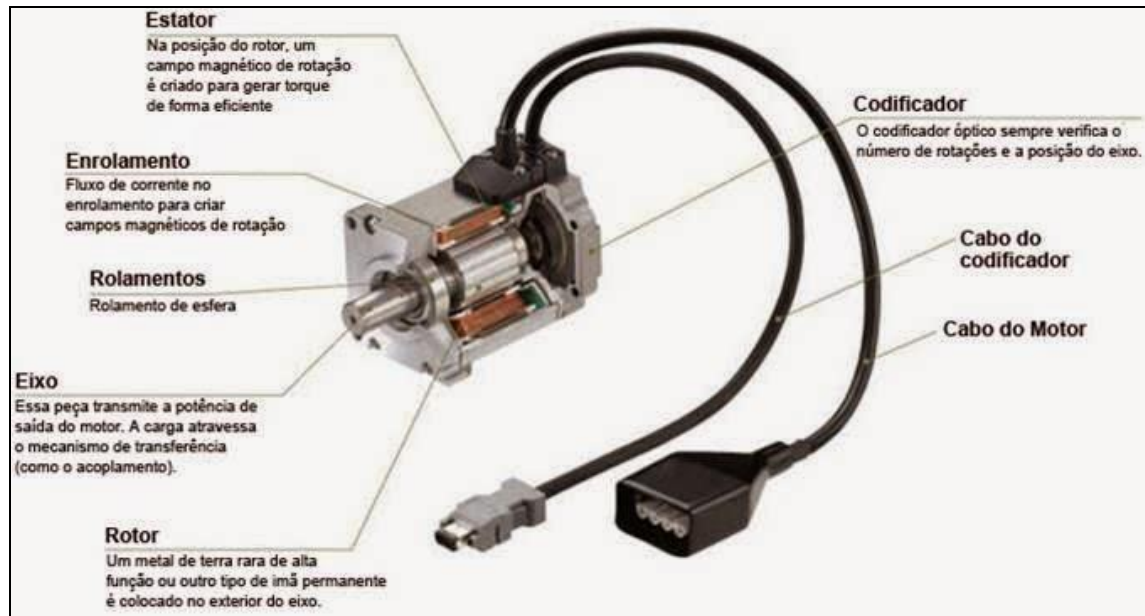


Figura 2.2 - Servomotor [5].

Devem ser considerados como parte do sistema de servomecanismo os dispositivos que auxiliam no controle de velocidade e posição, chamados de encoder e resolver. Dispositivos estes incorporados no eixo do servomotor responsáveis por enviar sinais elétricos ao servoconversor para efeito referenciamento.

2.2 ELEMENTOS ELÉTRICOS DE COMANDO

Em aplicações industriais existem componentes elétricos que são destinados ao envio de informações para o mecanismo de controle. São, geralmente, chaves mecânicas que mudam o estado físico e permitem a passagem/interrupção de corrente elétrica que são tratados como sinais de comando para sistemas embarcados. Os principais elementos de comando são:

Botão de acionamento: quando se aperta o botão, o elemento de comando atua contra a força da mola, interligando eletricamente a conexão NA (normalmente aberta) e desconectando a conexão NF (normalmente fechada). A mola é utilizada para voltar o botão a sua posição inicial.

Interruptor com trava: o interruptor com trava é um elemento que memoriza a sua condição, ou seja, se pressionado ele mudará de condição (de aberto para fechado ou de fechado para aberto) para que volte a condição inicial, é preciso pressioná-lo novamente. Isso ocorre porque o interruptor não possui um dispositivo de retorno automático.

Chaves fim de curso: são utilizadas para detectar o fim de um movimento mecânico ou de uma junta robótica por exemplo. Trata-se de uma chave eletromecânica convencional que opera somente em on/off e que apresenta duas formas gerais de operação, normalmente aberta ou normalmente fechada. Para este dispositivo é necessário que ocorra o contato mecânico entre as partes que estão em movimento para que a mola interna da chave fim de curso seja acionada e os contatos sejam atuados [6].

2.3 ELEMENTOS SENSORIAIS

Basicamente os elementos sensores podem ser denominados transdutores. Transdutor é todo dispositivo que fornece uma resposta de saída, da mesma espécie ou diferente, que reproduz certas características do sinal de entrada, a partir de uma relação definida. A maioria dos sensores é um transdutor elétrico, pois converte a grandeza de entrada característica da medida de fenômenos como som, luz ou pressão em uma grandeza elétrica, que pode ser medida por um circuito eletroeletrônico. As grandezas elétricas que apresentam variações proporcionais a grandezas que estão sendo medidas e indicadas pelos sensores são tensão, corrente e resistência elétrica [6]. Alguns dos sensores mais utilizados são descritos a seguir:

Fototransistor: é um sensor baseado no transistor cuja junção coletor-base fica exposta à luz e atua como um fotodiodo. O transistor amplifica a corrente e fornece alguns miliampères com alta luminosidade. A velocidade de resposta desse tipo de sensor é menor do que a do fotodiodo ainda que suas aplicações sejam as mesmas, exceto em sistemas de fibra ótica, que operam em alta frequência. Esse sensor pode ser utilizado para ligar ou desligar um motor, soar alarmes, abrir portas e etc.

Fotodiodo: é um sensor baseado num diodo semicondutor cuja junção também está exposta à luz. A energia luminosa desloca elétrons para a banda de condução, reduzindo a barreira de potencial pelo aumento do número de elétrons que podem circular se for aplicada uma polarização reversa. Esse tipo de sensor é usado em controles remotos, sistemas de fibra óptica, detectores ópticos de fumaça, leitoras de código de barras, scanners, canetas ópticas, etc.

Sensores ópticos: um sensor óptico é formado por um emissor e por um receptor de luz. O emissor de luz óptico pode ser um LED ou uma lâmpada. O receptor é um componente fotossensível, isto é, sensível à luz, como o fototransistor, os fotodiodo ou resistores variáveis pela luz. O princípio de funcionamento de um sensor óptico baseia-se num circuito oscilador que gera uma onda convertida em luz pelo emissor. Um objeto, quando aproximado do sensor óptico, reflete a luz do emissor para o receptor. Um circuito eletrônico identifica essa variação no circuito de alimentação e emite um sinal digital que pode ser utilizado para informação, controle ou inspeção. Caso o objeto seja muito escuro ou transparente a ponto de não conseguir emitir de volta a luz enviada pelo emissor, o sensor óptico não funcionará da maneira correta. Podem ser construídos em um único corpo ou também podem ser dividido em dois corpos, um para emitir a luz e outro apenas para recebê-lo, neste caso um objeto da cor preta seria detectado sem problema algum [6]. A imagem 2.3 mostra alguns sensores ópticos que são aplicados na robótica e automação industrial.



Figura 2.3 - Sensores ópticos [7].

Esse tipo de sensor é largamente utilizado na automação industrial, pois ele é dotado de diversas características positivas por não requererem contato mecânico para acionamento, não possuir partes móveis, apresentam pequenas dimensões, o seu chaveamento é mais seguro, são insensíveis à vibração e ao toque, possui várias possibilidades de configuração e podem ser blindados.

Sensores ultrassônicos: Seu princípio de funcionamento é muito simples, semelhante ao de um sonar utilizado em navios, onde uma onda sonora é emitida em uma determinada direção e quando encontrado um obstáculo, a onda é refletida de volta, nisso, conhecendo a velocidade da

onda mecânica e o tempo gasto para retorno tem-se a distancia que o aparelho receptor está do obstáculo. Assim um sensor ultrassônico funciona a partir da emissão de som em alta frequência, inaudível ao ser humano. O tempo de propagação é diretamente proporcional á distância do obstáculo a ser identificado. Estes sensores são geralmente utilizados como sensores de proximidade. Suas aplicações na indústria são para detecção de materiais, exceto aqueles que absorvem o som, leitura de nível de reservatórios e medição de peças em geral.

Sensores magnéticos de proximidade: São sensores compostos por materiais magnéticos onde, na passagem de um material ferromagnético, um ímã é atraído fechando o contato alimentando o comando elétrico. Para sensores mais modernos, essa identificação é realizada através da detecção da variação de um campo eletromagnético gerado pelo sensor. Este campo é variado justamente pela interferência de um material de ferromagnético. Esses sensores, muito utilizados para detectar fim de curso de sistemas automatizados, podem ser classificados como indutivos ou capacitivos. Os indutivos usam a alteração do campo magnético em uma região para detectar a presença de um objeto metálico, e os capacitivos, a alteração da capacitância em uma determinada região causada pela presença dos objetos não metálicos [6]. A figura 2.4 mostra os sensores de proximidade citado acima.



Figura 2.4 - Sensores de proximidade [8].

Potenciômetros de precisão: um potenciômetro é um sensor analógico que trabalha como um divisor de tensão. Um potenciômetro de precisão difere de um comum pela precisão e pela faixa de variação de resistência maior, ou seja, ele tem uma variação angular de muitas voltas. Trata-se de um dispositivo que transforma uma variável mecânica em uma tensão elétrica. Esses sensores são muito utilizados em malhas de controle de motores elétricos para a detecção de erro de posição e velocidade. Esses tipos de sensores apresentam um baixo custo de aquisição, podem sofrer alterações de temperatura, e fornecem informação de posição absoluta. Como aplicação é possível citar os manipuladores robóticos e as máquinas rotativas de corrente contínua [6].

Transformadores rotativos: Transformadores rotativos são transdutores que operam obedecendo ao mesmo princípio de atuação dos transformadores lineares, com características semelhantes, em que apenas a variável de entrada é um deslocamento angular. Tais transformadores são usados particularmente em sistemas em que seja necessário medir pequenos deslocamentos angulares, como lemes de direção em navios, aviões, válvulas hidráulicas e radares em que a rotação do eixo da antena é limitada. Sua principal vantagem em relação aos potenciômetros de precisão é a não existência de atritos e de ruído elétrico, provocado pelo cursor do potenciômetro.

Encoders ópticos: Encoders são sensores digitais comumente utilizados para fornecer a realimentação de posição em atuadores. São compostos por discos de vidro ou metal que giram entre uma fonte de luz e um par de fotodetectores. Assim, o disco é codificado com setores alternados de transparência e opacidade, gerando pulsos de luz e escuridão quando na rotação do disco. O sinal digital é fornecido em forma binária devido a configuração mecânica do disco e utilizado para fornecer informação para CLP's e microcontroladores. Existem basicamente dois tipos de encoders, os encoders absolutos e os incrementais.

Para os incrementais os sinais de saída é constituído de um trem de pulsos gerados a partir de LEDs provenientes de dois sensores defasados de 90° colocados no disco óptico, formando uma barreira, onde a interseção do disco entre os sensores gera os sinais que são emitidos para os dispositivos controladores.

Para os encoders absolutos o sistema de codificação é associado a cada posição angular do eixo medido. A principal vantagem desse tipo de encoder é a ausência de necessidade e inicialização de posição. Com frequência codificada em código Gray, os encoders absolutos são muito mais complexos e de custo mais elevados. Em outras palavras, o encoder incremental fornece um trem de pulsos proporcional a velocidade e um encoder absoluto possui um código específico em binário para cada ângulo que o eixo assume, nisso em uma possível falta de energia, o controlador saberá qual a posição angular do eixo [6].

2.4 SISTEMAS EMBARCADOS

2.4.1 Microprocessadores

Os microprocessadores foram incorporados e lançados primeiramente pela Intel Corporation que lançou no mercado uma calculadora com um dispositivo diferente de comando e controle chamado de processador, por ser constituído por apenas um único chip. Este chip possuía em torno de 3.000 transistores e em seguida as aplicações foram surgindo com o lançamento de novas calculadoras e computadores pessoais. Com o avanço tecnológico no ramo da microeletrônica outras características foram incorporadas ao longo das últimas décadas como unidades de gerenciamento de memória, memória cache, coprocessadores numéricos, dentre outros, tornando assim os microprocessadores cada vez mais complexos [9].

O microprocessador é um dispositivo lógico programável composto por elementos de silícios em um único chip desenvolvido através da tecnologia VLSI (circuito integrado em alta escala), conhecido como circuito integrado de alta escala. Este microchip é provido de memória, lógica booleana, operações aritméticas via hardware e software para receber diversos tipos de programação e conseguir se comunicar com dispositivos periféricos. Dentro do microprocessador encontram-se Contadores, Flip-flops, Codificadores, Circuitos de operações matemáticas, Registradores e Decodificadores, o que permite o dispositivo receber linhas de códigos de programação e realizar tomadas de decisões de acordo com o que a CPU está estipulada para executar.

Observando o seu funcionamento nota-se que o dispositivo lê as instruções armazenadas na memória, compila, processa, obtém seus operadores quando necessário, manipula os dados de acordo com o que foi programado e registra ou atua dependendo da instrução. Ele usa suas portas de entrada e saída de dados para que possa se comunicar com dispositivos externos. Essa função é muito importante, pois permite o microprocessador receber sinais externos e enviar comandos para executar ou não alguma função pré-determinada na sua rotina de programação [9].

A CPU do processador realiza todas as operações lógicas e aritméticas, nela estão localizados os registradores, contadores e unidades de controle em geral. Os *registradores* correspondem a uma memória local rápida que o microprocessador possui com o objetivo de armazenar os dados e instruções. A CPU possui dois tipos de registradores. Aqueles utilizados para operações lógicas e aritméticas, que são chamados de *registradores de propósito geral*, e aqueles registradores que possui funções já específicas e são chamados de *registradores especiais*.

Existe no processador uma unidade específica para trabalhar com portas lógicas e blocos de aritmética. A ALU (Unidade Lógica e Aritmética) implementa essas operações e geralmente armazena seus resultados no acumulador. A Unidade de Controle (UC) gerencia todas as funções

de um microprocessador onde, ela retira cada instrução da memória, processa o conteúdo decodificando as operações e fornece os sinais de controle necessários para a UC que em geral é constituída pelas seguintes partes:

Referencia de Tempo: esta parte é conhecida como Clock. É responsável por implementar o funcionamento síncrono do processador, indicando os instantes onde cada etapa da rotina deve ocorrer. O clock determina a velocidade de operação do microprocessador. Isso é possível porque o sinal do Clock é fornecido por um circuito oscilador á cristal que pode ser configurado de diversas maneiras. Quando utilizado um circuito oscilador RC a velocidade de operação não é grande e também não é muito precisa, mas para executar tarefas que não necessitem de precisão no “clock” e de grande velocidade de operação, esta opção é a mais simples geralmente utilizada. Para outra configuração, como o circuito Oscilador XT o microprocessador utiliza um cristal comum para realizar seu clock. Esta configuração é a mais utilizada por programadores que fazem uso de microcontroladores, sendo necessário apenas um cristal e dois capacitores cerâmicos, pois isso torna o manuseio mais versátil. Para aplicações mais avançadas são utilizados cristais ressonantes de alta frequência na configuração HS ou um circuito oscilador externo;

Controle e Decodificação de instrução: memória apenas de leitura que são utilizadas para a execução de cada linha de instrução da rotina. As instruções que estavam armazenadas na memória geram os códigos do programa que realizará a tarefa definida por ela.

2.4.1.1 Microcontroladores PIC

O PIC é um componente eletrônico pertencente à classe dos microcontroladores da arquitetura Harvard. No geral, um microcomputador pode possuir uma memória RAM, uma memória não volátil EEPROM, memória de programa, controladores de entradas e saídas digitais e analógicas em torno de uma CPU, dentro de um único chip. O PIC pode ser programado para executar diversas tarefas, como controlar um dispositivo eletromecânico, realizar medições, exibir informações em um display, ou simplesmente ascender alguns LEDs. É um equipamento simples e com um custo muito baixo [10].

A programação pode ser feita em um computador com softwares específicos, geralmente gratuitos e programados em Assembly ou Linguagem C. A maneira mais tradicional de fazer isso é colocando o PIC em um gravador apropriado para programar e posteriormente, o PIC é

transferido para o circuito definitivo, onde pode ser soldado diretamente à placa de circuito impresso ou inserido em outro soquete. Há vários tipos diferentes de programadores cujo projeto pode ser encontrado na Internet ou adquirido pronto de determinados fornecedores. A maioria deles é de fácil construção e podem ser conectados a porta serial do PC. De fato, qualquer computador que disponha de uma dessas interfaces pode ser usado para programar o PIC. A título ilustrativo, a figura 2.5 mostra a imagem de um Microcontrolador muito utilizado em protótipos e projetos de sistemas embarcados.



Figura 2.5 - PIC 16F877A.

2.4.1.2 Plataforma Arduino

O Arduino é uma plataforma aberta que foi desenvolvida para facilitar e agilizar os projetos que envolvam microcontroladores. A plataforma Arduino é capaz de ler entradas analógicas ou digitais para receber sinais de sensores, pulsos através de botões ou ler mensagens e transferir esses sinais para as suas saídas ativando motores, ligando LEDs e até mesmo trabalhar de forma online em rede utilizando protocolos e placas *boards* específicas. É possível programar uma rotina para que ele execute o que foi especificado para executar através do Microcontrolador instalado na placa. Para tanto, é necessário utilizar a linguagem de programação específica do Arduino e seu próprio software de programação e de upload.

No decorrer dos anos, o Arduino tem sido bastante utilizado em diversos projetos de desenvolvimento tecnológico, sendo que a cada dia, desenvolve projetos cada vez mais complexos. O Arduino nasceu na Itália com o propósito de ser uma ferramenta rápida focada em estudantes sem muito conhecimento em programação e eletrônica.

Para trabalhos envolvendo microcontroladores esta placa é muito aceita devido ao baixo custo e a praticidade de manuseio. Sua interface é articulada para trabalhar com programadores

iniciantes e programadores profissionais podendo ser utilizado em projetos avançados para construir protótipos de engenharia, música, automação, robótica, etc. Ou seja, qualquer estudante ou profissional consegue fazer uso desta tecnologia. Existem outras diversas plataformas como *Parallax Basic Stamp*, *Netmedia's BX-24*, *Phidgets*, *MIT's Handyboard*, dentre outros.

Como foi mencionado, existem diversas plataformas para milhares de aplicações diferentes, várias placas, vários modelos em que cada versão possui um número de portas equivalente a sua aplicação. A tabela 2.1 apresenta alguns modelos de plataformas Arduino.

Tabela 2.1 - Modelos de mercado [28].

Modelo	Microcontrolador utilizado
<i>Arduino Duemilínove</i>	ATmega168 ou ATmega328
<i>Arduino Diecimília</i>	ATmega168
<i>Arduino Mega</i>	ATmega1280
<i>Arduino Nano</i>	ATmega168 ou ATmega328
<i>LilyPad</i>	ATmega168V
<i>Pro</i>	ATmega168 ou ATmega328
<i>Pro mini</i>	ATmega168

Na figura 2.6 é possível visualizar algumas plataformas para exemplo:

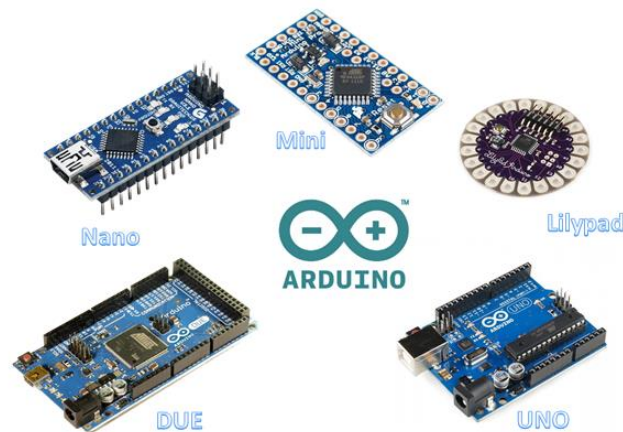


Figura 2.6 - Diversidade Arduinos [11].

Uma das vantagens do Arduino em relação ao PIC é a praticidade e a facilidade de manuseio e programação. Para o PIC deve ser considerado todo um circuito eletrônico para que o funcionamento do Microcontrolador ocorra de maneira correta. É necessário dimensionar e utilizar um Clock para referência de tempo e sincronização, resistores e capacitores para compor o circuito de hardware e de um circuito separado para alimentar toda a parte eletrônica do

dispositivo. Enquanto isso o Arduino não exige que o programador realize essa tarefa, todo o circuito necessário para trabalhar com ATmega já está inserido na placa de desenvolvimento do Arduino juntamente com o circuito de alimentação dos dispositivos eletrônicos. Outra grande vantagem é que o circuito de gravação de dados do Arduino também já está incorporada na plataforma, evitando a necessidade de retirar os Microcontrolador da placa e transferir de um circuito para o outro, uma manobra realizada constantemente utilizando o PIC.

Arduino Due é a primeira placa de desenvolvimento Arduino baseada em ARM. Essa placa possui um Microcontrolador *ARM CORTEX M3 32bit* programável através da IDE do Arduino, ou seja, essa placa possui uma capacidade computacional muito superior aos outros Arduinos com a vantagem de poder utilizar a mesma linguagem de programação dos outros Arduinos [11].

O Arduino Due tem 54 entradas/saídas digitais, das quais 12 podem ser utilizadas como saídas PWM, 12 entradas/saídas analógicas, 4 UARTs (portas serias de hardware), clock de 84MHz, capacidade de conexão USB-OTG, 2 DAC (digital para analógico), 2 TWI, conector para alimentação, pinos SPI, pinos JTAG, botão de reset e botão para apagar o código. Devido a limitações de tensão impostas pelo *Atmel SAM3X8E*, o Arduino Due funciona com 3,3V, que é a máxima tensão que os pinos de entrada/saída podem tolerar.

Dados físicos do dispositivo:

- Tensão de operação: 3,3V;
- Tensão de alimentação: 6-20V;
- Pinos de entradas/saídas digitais: 54 (dos quais 12 possuem saída PWM);
- Pinos de entrada analógica: 12;
- Corrente máxima em todas as entradas/saídas: 130mA;
- Corrente máxima no pino de 3,3V: 800mA;
- Corrente máxima no pino de 5V: 800mA;
- Memória Flash: 512 KB;
- Clock: 84MHz.

2.5 CIRCUITO PONTE H

Uma ponte H trata-se de uma configuração entre chaves elétricas que permitem acionamento de um motor de corrente contínua através de sinais gerados por um

microcontrolador ou outro dispositivo embarcado. Devido a disposição de seus componentes torna-se muito fácil inverter o sentido de giro de um motor DC apenas mudando os terminais de controle que são usados para acionar as chaves. Em circuitos digitais a sua utilização é extremamente necessária porque os microcontroladores não conseguem fornecer corrente suficiente para alimentar a carga nem mesmo trabalhar no mesmo nível de tensão, nisso, é necessário utilizar outros componentes que possuem essa capacidade.

Se for conectado uma fonte de corrente contínua em um motor DC será observado que a velocidade da máquina será constante e que o motor funcionará em um único sentido de rotação. Caso seja necessário alterar o sentido de rotação do motor, basta inverter os pólos de alimentação da fonte para que ocorra a inversão. A ponte H permite fazer esta operação de inversão de sentido de rotação sem que seja necessário realizar esta operação manualmente.

Uma ponte H simples é composta por relés eletromecânicos ou dispositivos semicondutores posicionadas em formação “H”, onde o motor se localiza ao centro do circuito e as chaves em suas extremidades. Para aplicações que exigem controle de velocidade, os dispositivos de chaveamento são compostos por transistores, IGBT's ou MOSFET's. A figura 2.7 ilustra o circuito eletrônico de uma ponte H utilizando transistores.

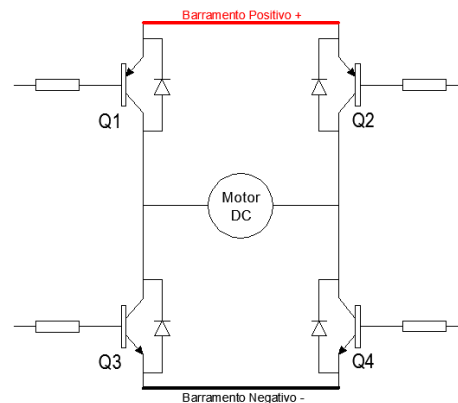


Figura 2.7 - Configuração Ponte H.

Para acionar o motor é necessário ligar as duas chaves opostas diagonalmente (Q1 e Q4) fazendo com que a corrente elétrica flua do polo positivo para o negativo, consequentemente fazendo o motor girar em uma direção. Para inverter o sentido, basta desligar as chaves que estavam ligadas e acionar as outras duas (Q2 e Q3) para que o fluxo ocorra de forma reversa fazendo o motor rodar no sentido contrário. A figura 2.8 exibe o sentido da corrente em relação aos semicondutores polarizados.

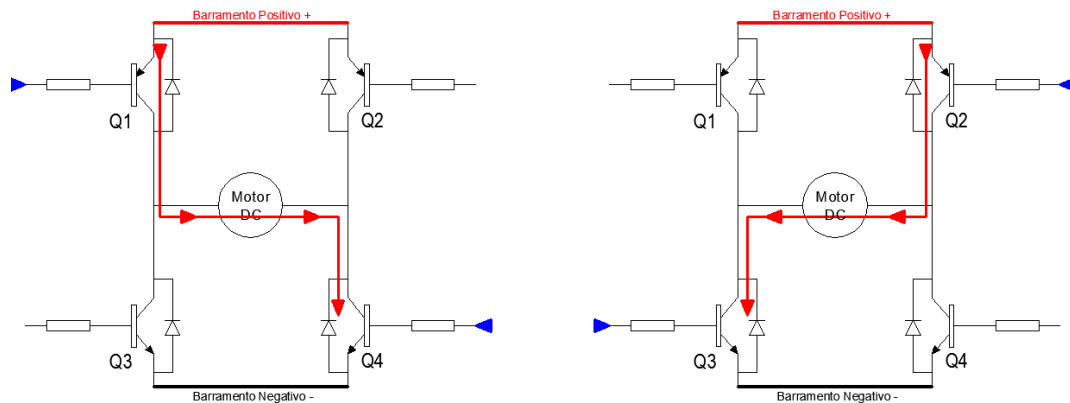


Figura 2.8 - Circuito para acionamento da ponte H.

Um cuidado muito importante deve ser tomado em relação ao acionamento das chaves da ponte H, pois observando o circuito verifica-se que ao acionar os transistores Q1 e Q3 ou Q2 e Q4 ao mesmo tempo existe uma circulação direta de fluxo de corrente do polo positivo para o negativo realizando a formação de um curto-circuito podendo danificar os componentes e todo o sistema eletroeletrônico.

Destaca-se que os transistores da ponte H atuam na região de corte/saturação.

2.6 SISTEMAS MECATRÔNICOS

Basicamente a mecânica tem como objetivo o atendimento de uma necessidade humana, com vistas a proporcionar conforto, bem-estar e segurança. No entanto essas necessidades não são fixas e exigem constantes adaptações dos sistemas mecânicos. Neste aspecto, pode-se identificar duas concepções básicas a serem adotadas no projeto de um sistema mecânico, o *sistema mecânico passivo* e o *sistema mecânico ativo* [6].

2.6.1 Sistemas Mecânicos Passivos

A forma geométrica do sistema e de cada componente deve ser detalhadamente mensurada e especificada. Relaciona-se a formas macroscópicas, como comprimento, altura, largura etc., e até detalhes como acabamento superficial. A mecânica possui conceitos como cinética, dinâmica, resistência de materiais e ciência de materiais, e de um processo de criação, o

projetista define a forma e a maneira de fabricação do sistema mecânico, de modo que ele execute as funções exigidas. Algumas das ciências básicas da mecânica estão relacionadas a seguir:

Cinemática: define a relação entre a geometria e a grandeza física a ser gerada como posição, velocidade, aceleração, força etc.

Dinâmica: garante que as partes que compõem o dispositivo, cada uma possuindo uma inércia, executem as funções requeridas segundo uma ordem cronológica desejável.

Resistência dos Materiais: garante que as relações geométricas mesmo diante de deformações provocadas por esforços ou de rompimentos completos do componente, o que é uma forma extrema de deformação.

Termodinâmica: define as dimensões e a geometria dos componentes do mecanismo, a fim de garantir que elas não atinjam temperaturas que possam comprometer a resistência mecânica.

Mecânica dos fluidos: define esforços gerados por líquidos em repouso ou em movimento e, por meio desses esforços, a geometria e as dimensões dos componentes.

Economia: leva em conta a existência de uma forte relação entre o custo e a forma geométrica a ser obtida. Como a tendência é minimizar o custo ele acaba contribuindo na definição da forma geométrica.

É possível enxergar através desta breve definição que todas essas ciências mencionadas conduzem a definição da forma geométrica do equipamento. É possível também afirmar que a função executada por um sistema mecânico passivo fica definida assim que se escolhe a sua forma e seu material, ou seja, assim que ele é projetado. Essa é uma característica relevante dos sistemas passivos.

Existem casos em que o sistema mecânico já prevê uma variedade de necessidades e, em função disso, o projeto é realizado de modo que um mesmo sistema incorpore vários subsistemas, cada um destinado a atender um tipo de necessidade. Um exemplo muito tradicional é a caixa de transmissão de um carro que é projetado para atender vários tipos de velocidade. Apesar de não requer nesse caso um novo projeto ou uma nova construção, o processo de adaptação a uma nova necessidade é integralmente dependente da intervenção humana. Esta é mais uma característica dos sistemas passivos. Caso seja desejável mudar a concepção ou os movimentos do protótipo mecânico é necessário refazer todo o projeto e construir um novo equipamento [6]. O figura 2.9 esboça o fluxo de projeto de um sistema passivo.



Figura 2.9 - Metodologia de concepção de sistemas mecânicos passivos [6].

2.6.2 Sistemas Mecânicos Ativos

Muitas vezes, o modelo de sistema mecânico passivo pode sofrer algumas alterações, nesse caso, o projeto prevê a inclusão de um elemento de ajuste, cuja função é permitir que o sistema já projetado e construído tenha seu desempenho funcional alterado. Mas a grande diferença do modelo ativo para o passivo é a inclusão do conceito de controle automático na etapa de concepção do sistema mecânico. O conceito de controle automático é definido através da teoria em que o elemento de ajuste realiza as devidas correções de parâmetros internos do sistema, a fim de obter a saída desejada. Lembrando que o elemento de ajuste atua sem a intervenção humana.

Um exemplo de sistema mecânico ativo é a locomotiva equipada com um regulador automático de velocidade, proposta por James Watt em 1978. Nesse regulador, assim que a velocidade de rotação da locomotiva aumenta, a força centrífuga que age sobre os contrapesos também aumenta, fazendo com que o deslizador suba. Isso, por sua vez, faz com que uma válvula se feche e diminua a quantidade de vapor que flui para a locomotiva. A velocidade da locomotiva tende então a cair, o que resulta no oposto de um sistema passivo [6].

O acelerador de um carro é um sistema passivo devido a necessidade da intervenção humana sobre os pedais de aceleração. Em carros modernos já são implementados controle de velocidade automático, caracterizando neste caso um sistema ativo com realimentação elétrica.

Os problemas relativos aos sistemas mecânicos passivos são solucionados nos sistemas mecânicos ativos, os quais viabilizam a adaptação em tempo real às alterações das funções exigidas pelo sistema. Ao mesmo tempo, controle automático aumenta a precisão com que as necessidades são atendidas. Os sistemas mecânicos dotados dessas vantagens permitiram a criação de máquinas capazes de executar funções impossíveis de ser viabilizadas na forma de sistema passivo [6].

2.6.3 Sistemas Ativos com Realimentação Elétrica

A partir de 1950 foi inventado o transistor, marcando o início de uma nova era de notável desenvolvimento da tecnologia eletrônica, como surgimento de novos sensores, atuadores e do computador. Em particular, o computador representou um avanço no que diz respeito a fazer interpretações, análise e tomadas de decisões, tarefas até então exclusivas do ser humano. Esse desenvolvimento acarretou, entre outros avanços tecnológicos, o desenvolvimento da tecnologia de controle por meios eletrônicos, ou seja, por intermédio da utilização de sensores, atuadores e controladores implementados em circuitos analógicos e/ou digitais. A nova tecnologia foi rapidamente incorporada a engenharia mecânica, dando origem a uma nova classe de sistemas mecânicos ativos com realimentação elétrica denominada sistemas mecatrônicos.

Em meio aos avanços da tecnologia, surgiram os conversores de sinais, que convertem um sinal mecânico em sinal elétrico (sensores) ou um sinal elétrico em sinal mecânico (atuadores), bem como os dispositivos eletrônicos lógicos digitais com capacidade de tratamento de sinais elétricos. A implementação dos sensores e dos atuadores em sistemas mecânicos podem acarretar em imprecisões construtivas, devido as folgas e aos atritos nas diversas partes móveis do sistema, comprometendo a precisão e o comportamento do projeto.

O tratamento de sinais é uma grande ferramenta nas técnicas de controle permitindo a comunicação e o envio e recebimento de dados sobre o processo. Os dispositivos eletrônicos apresentam a vantagem de permitir a execução de ampla variedade de operações matemáticas com precisão elevada a uma velocidade também elevada quando comparada com os sensores mecânicos.

A utilização de sensores na realimentação do sistema traz diversos benefícios devido a formação de um sistema mais eficiente e confiável em termos de:

- Precisão de Controle;
- Tempo de resposta;
- Capacidade de realizar algoritmos de controle de elevada complexidade.

O sistema de controle utilizado em sistemas com realimentação elétrica é apresentado no diagrama da figura 2.10.

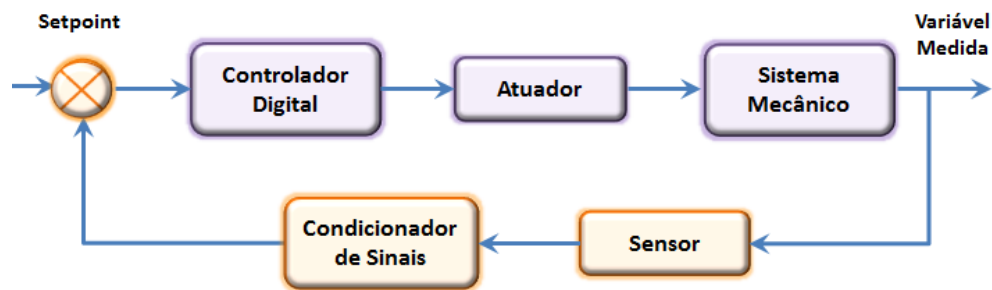


Figura 2.10 - Estrutura de um sistema mecatrônico [6].

2.7 ASPECTOS CONSTRUTIVOS DE MANIPULADORES

Um robô é um sistema mecânico de geometria variada composto por corpos rígidos articulados entre si destinados a sustentar e posicionar uma ferramenta terminal, que, dotado de garra mecânica ou ferramenta especializada, fica em contato direto com o processo produtivo.

Um braço robótico executa movimentos no espaço transportando peças, caixas, ferramenta, etc., e é controlado por uma CPU através de sensores espalhados no processo. Todo braço robótico é composto por juntas que permitem o movimento relativo entre as articulações e quase sempre possuem uma base fixa, onde o primeiro vínculo está preso a ela. A sua mobilidade depende do número de vínculos e juntas anexados em sua estrutura.

2.7.1 Juntas Robóticas

O braço manipulador é capaz de se movimentar através das articulações, também denominadas de eixos e pode ser linear ou rotacional. O número de juntas determina o grau de

liberdade do robô. A maioria deles possui de três a seis eixos nos quais podem ser definidos em duas classes: eixo do corpo e eixo da extremidade do robô. Os eixos da base do robô são chamados de cintura, OMBRO e COTOVELO e os eixos das extremidades são chamados de rolamento, mergulho e guinada, que possibilita sua orientação. Na maioria dos casos os robôs industriais possuem seis graus de liberdade onde três delas são destinadas para o controle de posicionamento da ferramenta e as outras três destinadas para a orientação do conjunto no espaço físico no qual está instalado [6].

As juntas podem ser caracterizadas como:

- *Deslizantes*: que permitem o movimento linear entre dois vínculos. São compostas de dois vínculos alinhados a um dentro do outro, o vínculo interno escorrega pelo externo e dá origem ao movimento linear.
- *Rotativas*: cuja à conexão possibilita movimentos de rotação entre dois vínculos unidos por uma dobradiça comum, com uma parte podendo se mover num movimento cadenciado em relação á outra. As juntas rotativas são utilizadas em diversas ferramentas e dispositivos, tais como em tesouras, limpadores para-brisas, etc.
- *Bola-e-encaixe*: conexão que se comporta como uma combinação de três juntas de rotação, permitindo movimentos de rotação em torno dos três eixos [6].

O número de articulações está associado ao número de graus de liberdade que o robô possui, isto é, para um movimento relativo em um único eixo, a articulação tem um grau de liberdade. Quando o movimento se dá em mais de um eixo, a articulação apresenta dois graus de liberdade. O ser humano, por exemplo, possui sete graus de liberdade contando do OMBRO até o punho.

A figura 2.11 representa a forma construtiva de uma junta robótica interligando dois vínculos formando uma articulação.

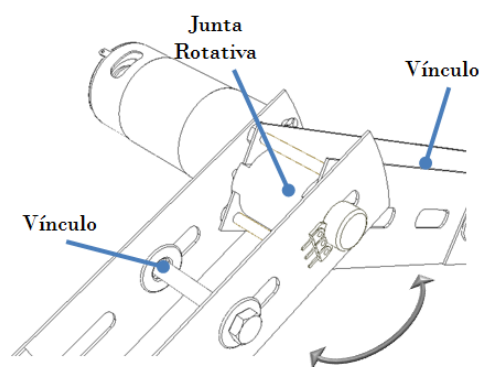


Figura 2.11 - Junta e Vínculos.

2.7.2 Vínculos

Três fatores importantes na construção dos vínculos (links) são a carga suportada, o peso do próprio braço e o grau de rigidez desse mesmo braço. Um braço pesado necessita de um motor mais robusto, o que torna o custo do equipamento muito alto. Um braço de baixa rigidez reduz a precisão do robô devido as vibrações e resposta à tensão. Para aumentar a rigidez mecânica do braço sem aumentar seu peso, frequentemente é usada uma estrutura oca, com maior rigidez mecânica se comparada com uma construção maciça que utiliza a mesma massa de material. É possível observar a ilustração dos vínculos de um manipulador na figura 2.12.

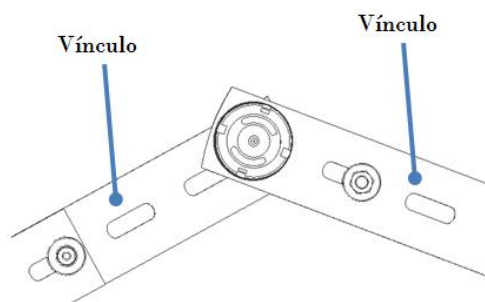


Figura 2.12 - Vínculos.

2.7.3 Garras e Ferramentas

Os robôs são projetados para uma finalidade específica, logo é necessário dimensionar a ferramenta que irá manusear as peças que serão trabalhadas ou movimentadas. Para as garras ou mãos mecânicas existem diversos tipos, como:

- Sujeição por pressão;
- Sujeição magnética;
- Sujeição a vácuo;
- Sujeição com peças a temperaturas elevadas;
- Resistentes a produtos corrosivos / perigosos;
- Com sensores;

- Pistolas pulverizadoras;
- Soldagem por resistência por pontos;
- Soldagem por arco;
- Dispositivos de furação;
- Polidoras;
- Dentre outros.

Uma garra pode ser comparável com uma mão humana. No entanto, ela não é capaz de simular todos os seus movimentos, fato que resulta na limitação dos movimentos a uma faixa de operações. A grande demanda por esse tipo de atuador acarretou o desenvolvimento de garras que podem manusear objetos de diferentes tamanhos, formas e materiais. Elas podem ser classificadas de várias formas:

- Garra de dois dedos;
- Garra de três dedos;
- Garra para objetos cilíndricos;
- Garra para objetos frágeis;
- Garra articulada;
- Garra a vácuo e eletromagnético;
- Adaptador automatizado de garras [6].

A figura 2.13 é um exemplo de garra de dois dedos.

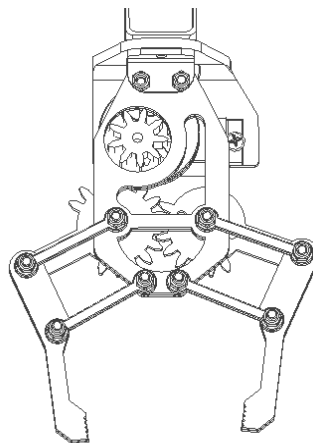


Figura 2.13 - Garra de dois dedos.

2.7.4 Precisão e Repetição

Dois parâmetros característicos dos robôs são a precisão e a capacidade de repetição. Por **precisão** entende-se a capacidade do robô de ir a uma posição destinada em um sistema de referência fixo, com um erro determinado. Trata-se de precisão em posicionamento absoluto. A capacidade de **repetição** pode ser entendida como a capacidade do robô de, uma vez conhecida e alcançada uma posição, e partindo sempre da mesma condição inicial, voltar a ir, isto é, repetir a posição com um erro determinado. A maioria dos manuais dos robôs fornecem informações sobre a capacidade de repetição, mas não sobre a precisão absoluta, muito mais difícil de obter [6].

2.7.5 Acoplamentos Mecânicos

Basicamente existem duas formas de acionamento de cargas através de eixos rotativos, os acionamentos diretos e indiretos. No caso do acionamento direto, o motor é montado com o eixo acoplado diretamente à junta rotativa do robô que vai realizar o movimento. Se for montado longe da junta ou próximo da base, o acionamento é indireto, e para esses casos se faz necessário o uso de elementos de transmissão como correntes, correias, cremalheiras, diferenciais e engrenagens.

Vantagens do acionamento indireto:

- Redução do peso do braço mecânico;
- Possibilidade de mudança na velocidade de rotação das juntas.

Desvantagens do acionamento indireto:

- Falta de precisão da operação da junta devido à liberdade mecânica dos pontos de conexão entre os dispositivos de transferência [6];

2.7.6 Formas de Acionamento dos Manipuladores

Os métodos de acionamentos de manipuladores robóticos podem ser divididos em três tipos diferentes, acionamento elétrico, acionamento hidráulico e acionamento pneumático.

Acionamento elétrico: para este tipo de acionamento são utilizados motores de passo, motores de corrente contínua e as vezes motores de corrente alternada. Na atualidade a utilização de servomecanismos são mais frequentes e possuem melhor precisão e simplicidade de controle. Neste tipo de acionamento, o controle é mais preciso, a estrutura é simples com fácil manutenção, a fonte de energia é acessível e o custo é relativamente baixo.

Acionamento hidráulico: esse tipo de sistema se faz necessário a utilização de unidades hidráulicas como fonte de força e trabalho. Esta unidade é composta por um motor elétrico e de um cilindro para realização de movimentos deslizantes. Essa unidade de acionamento provoca movimento em pistões que comprimem o óleo e é controlado por válvulas que regulam a pressão de óleo nas duas partes do cilindro e que impulsionam o pistão. Este sistema tem como vantagem um alto e constante momento sob uma grande faixa de variação de velocidade, precisão da operação e possibilidade de manter um alto momento por um longo período de tempo, quando parado. Acionamentos hidráulicos possuem fontes caras de energia, uma manutenção cara e intensa, exigem válvulas de precisão e existe a possibilidade de vazamentos de óleo.

Acionamento pneumático: é similar ao hidráulico, pois é composto por motores pneumáticos de movimento rotativo e cilindros pneumáticos de movimento deslizante. Neste caso é usado o próprio ar como meio de realização de trabalho e não o óleo. As vantagens de se utilizar este sistema é que os movimentos são rápidos durante a operação, o custo é relativamente pequeno, exige um baixo nível de manutenção, consegue manter um momento constante e uma grande faixa de velocidade e mantém um alto momento por longos períodos de tempo sem danos, quando parado. Os sistemas pneumáticos tem como desvantagens uma baixa precisão e possui momentos de vibração quando o motor ou o cilindro é parado [6].

2.8 CLASSIFICAÇÃO DOS ROBOS

Os robôs industriais podem ser classificados de acordo com os seus aspectos construtivos levando em consideração as disposições das juntas, o formato geométrico do robô, a forma de acionamento e o tipo de controle empregado. Geralmente são empregados nomenclaturas baseando-se no formato das três primeiras juntas mais próximas da base.

Os graus de liberdade de um robô podem ser obtidos de várias formas dependendo das combinações das juntas que podem ser lineares ou rotacionais. O código utilizado para a

classificação das juntas é a letra R, para indicar revolução e a letra P (*Prismatic*), para indicar movimentos deslizantes ou lineares.

2.8.1 Coordenadas Cartesianas

Os robôs de coordenadas cartesianas realizam movimentos lineares através de deslocamentos horizontais, verticais e em linha reta. Nessa configuração o robô possui três articulações deslizantes compondo o sistema PPP. São caracterizados pela sua pequena área de trabalho e seu alto grau de rigidez mecânica [6]. A figura 2.14 mostra a configuração geométrica pra um robô topo cartesiano.

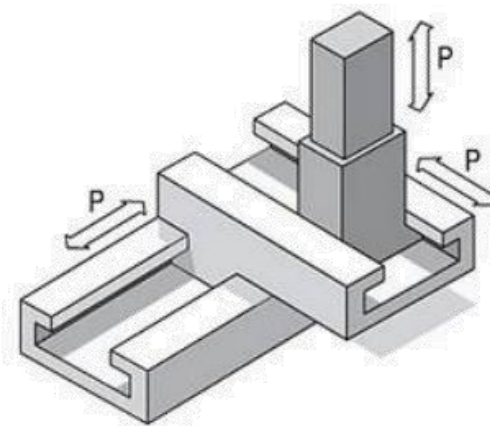


Figura 2.14 - Robô tipo cartesiano [6].

2.8.2 Coordenadas Cilíndricas

Um robô caracterizado como cilíndrico combina movimentos rotacionais com movimentos lineares. Normalmente, o eixo de rotação se encontra na base do protótipo para fazer a função de cintura e os outros movimentos são realizados de forma deslizante. Possui uma área de trabalho maior que a do robô cartesiano e uma pequena inferioridade na rigidez mecânica [6]. Seus graus de liberdades são compostos de forma RPP, assim como ilustrado na figura 2.15.

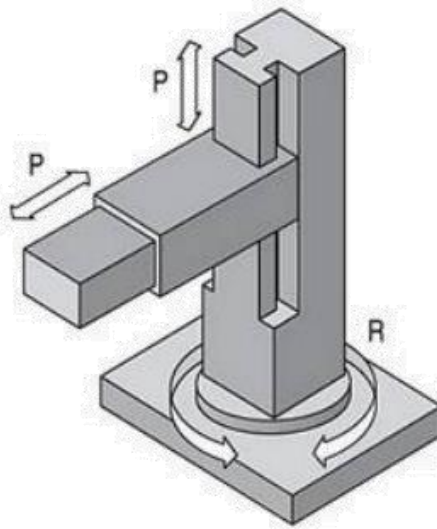


Figura 2.15 - Robô cilíndrico [6].

2.8.3 Coordenadas Polares (Esféricos)

Os robôs classificados como polares possuem uma articulação rotativa a mais que os robôs cilíndricos. Os polares são constituídos, normalmente, por duas juntas rotacionais e uma linear formando uma sequência RRP para compor os graus de liberdade pertencentes a essa configuração. Quando comparado com os robôs cilíndricos, os robôs polares ou esféricos possuem uma baixa rigidez mecânica, porém uma maior área de atuação [6]. A figura 2.16 mostra o formato característico do robô mencionado neste tópico.

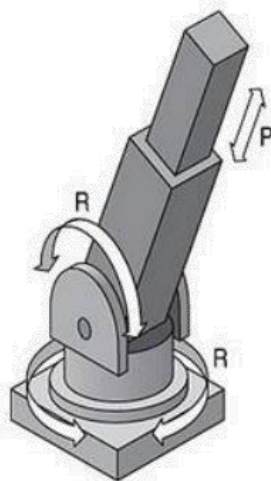


Figura 2.16 - Robô polar [6].

2.8.4 Coordenadas de Revolução (Articulado)

As juntas e os movimentos de um robô articulado se assemelham com o formato de um braço humano. É um dos robôs mais utilizados nas indústrias automobilísticas e possui graus de liberdade do tipo RRR devido a utilização restrita de juntas rotativas para compor as articulações. A sua área de atuação é maior do que qualquer outro tipo de robô, contudo apresenta uma baixa rigidez mecânica. Também é um dos robôs mais complicados de se controlar em função das juntas de revolução existentes. A figura 2.17 mostra os aspectos construtivos de um robô articulado.

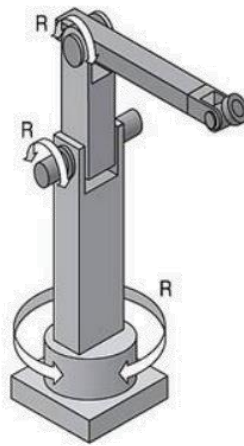


Figura 2.17 - Robô articulado [6].

2.9 SISTEMAS SUPERVISÓRIOS

Os sistemas de controle e comando nas indústrias vêm utilizando a automação como ferramenta para conseguir melhores níveis de flexibilidade durante o processo. Estão espalhados diversos controladores lógicos programáveis (CLP) conectados através de diversas redes diferentes e usualmente interligados com um sistema de supervisão. Devido a vasta gama de arquitetura e meios diferentes de se construir uma rede de comunicação industrial fica cada vez mais difícil escolher uma arquitetura de comando adequada aos sistemas distribuídos, sobretudo no que diz respeito a análise de tempo de resposta do sistema como um todo.

Um sistema de supervisão é responsável pelo monitoramento de variáveis de controle de uma máquina ou de um processo com o objetivo de fornecer subsídios para o operador controlar ou monitorar um processo automatizado mais rapidamente, permitindo a leitura das variáveis em tempo real e gerenciamento da planta.

2.9.1 Sistema SCADA (*Supervisory Control and Data Acquisition*)

Os sistemas de supervisão podem ser definidos como uma interface homem-máquina (IHM) amigável os quais utilizam tecnologias de computação e comunicação que permitem a supervisão e controle de sistemas automatizados, efetuando a aquisição dos dados em ambientes complexos, algumas vezes dispersos geograficamente. Com o desenvolvimento de softwares e o crescimento da demanda por sistemas supervisórios a disponibilidade e segurança da informação assumem elevada relevância, tornando-se necessário garantir que a informação esteja disponível e segura quando necessário, independentemente da localização do equipamento ou instrumento, portanto, é fundamental implementar mecanismos de acessibilidade, mecanismos de segurança e tolerância a falhas.

Atualmente, as principais empresas de desenvolvimento de sistemas supervisório são:

- Wizcon – Emation;
- Ifix – Intelluton;
- Intouch;
- Elipse;
- Cimplicity – Ge Fanuc;
- ActionView;
- Unisoft;
- National Instruments (LabVIEW).

A figura 2.18 apresenta o fluxo de informações quando adotado um sistema de supervisionamento.



Figura 2.18 - Fluxo de comunicação em supervisórios [12].

Principais características de um sistema supervisório:

- A interface deve ser de fácil compreensão para que o operador consiga entender todos os pontos do processo via tela de supervisório;
- Geração automática de relatórios com o controle estatístico do sistema;
- Histórico de tendências;
- Facilidade para interação com outros aplicativos;
- Acesso automático a banco de dados;
- Acesso compartilhado e remoto;
- Conexão em rede e por meio de modem ou rádio;
- Gerenciamento das condições de alarme.

A figura 2.19 mostra uma tela de supervisório onde é possível enxergar o volume dos tanques, o controle para acionamento de bombas e válvulas, monitorar a temperatura sistema e obter histórico do processo.

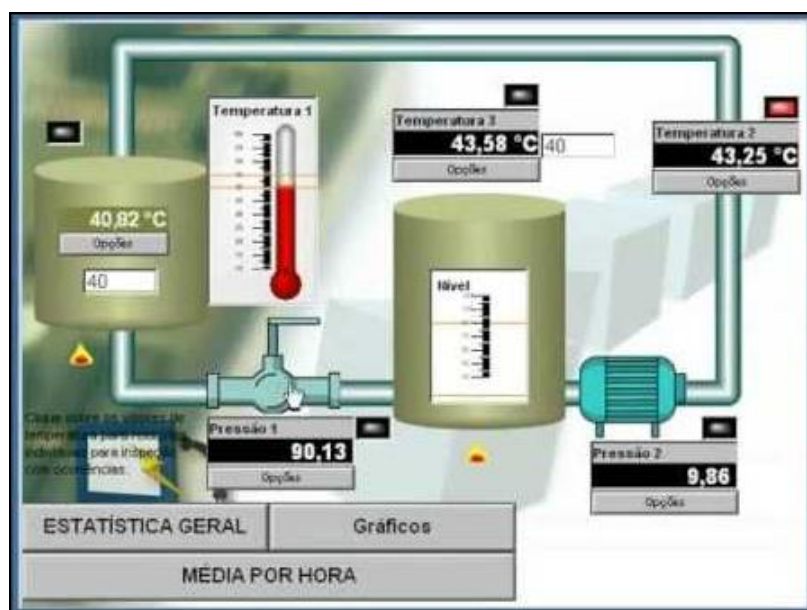


Figura 2.19 - Exemplo de tela de supervisório [13].

2.9.2 LabVIEW

LabVIEW é um ambiente de desenvolvimento criado especificamente para acelerar a produtividade de engenheiros e cientistas. Com sua sintaxe de programação gráfica, torna-se simples visualizar, criar e desenvolver códigos para sistemas de engenharia. É um software desenvolvido pela *National Instruments* com o intuito de aplicar em desenvolvimentos de equipamentos e monitoramento de processo fazendo o papel de supervisor. O programa utiliza programação baseada em fluxo de dados, que determina a o sentido de execução. O LabVIEW possui uma lista de objetos para que o usuário possa criar sua própria interface na tela chamada de painel frontal. É possível adicionar o código utilizando representações gráficas de funções para controlar os objetos deste painel [15].

O software é capaz de comunicar com dispositivos embarcados via comunicações:

- GPIB;
- VXI;
- Ethernet;
- RS-232;
- RS-485;
- USB;

Utilizando o LabVIEW, é possível criar aplicações de teste e medição, aquisição de dados, controle de instrumento, registro de dados, análise de medição e geração de relatório [14]. A figura 2.20 mostra uma tela que pode ser criada utilizando o software citado, chamada de tela frontal, e a figura 2.21 ilustra a programação em diagrama de blocos que está por trás da tela visualização (frontal).

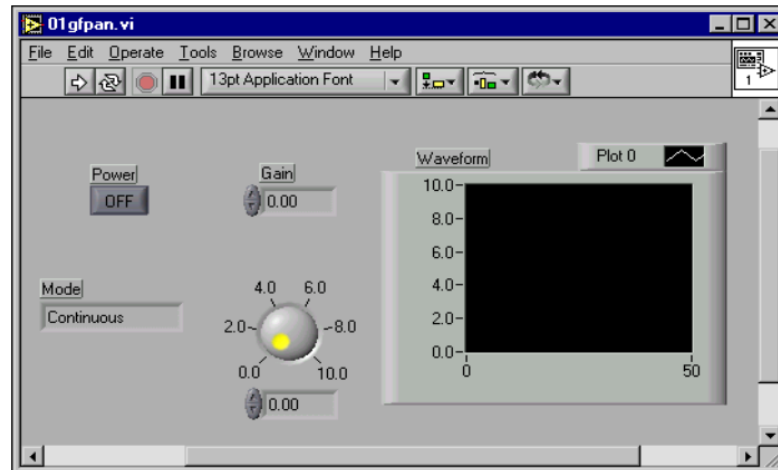


Figura 2.20 - Exemplo tela frontal LabVIEW [15].

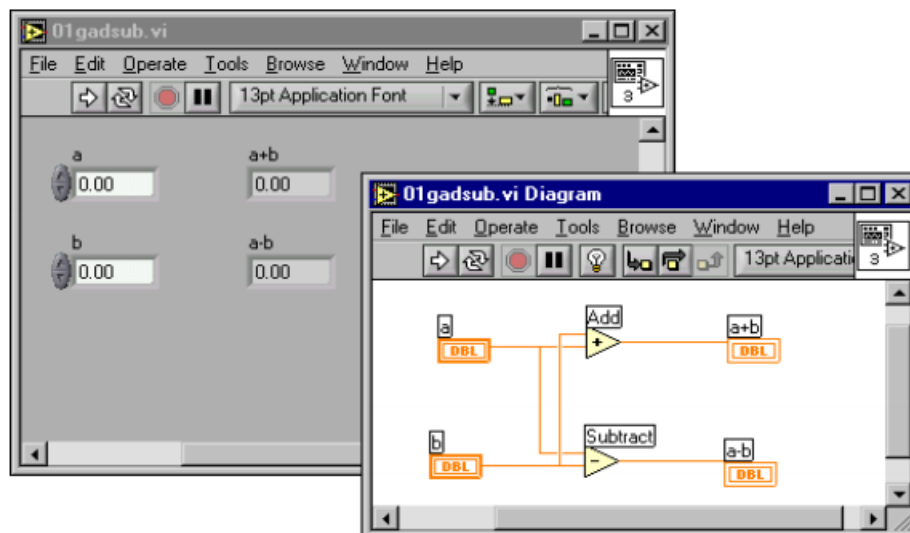


Figura 2.21 - Diagrama de bloco e painel frontal [15].

CAPÍTULO 3

DESENVOLVIMENTO DO PROJETO

Como mencionado anteriormente, o projeto consiste em construir um braço robótico com quatro graus de liberdade acionado com microcontrolador e supervisionado com um aplicativo supervisão. O braço mecânico deve ser capaz de pegar automaticamente objetos em uma esteira transportadora, identificar e separar os objetos metálicos e não metálicos, guardando-os em compartimentos diferentes. A identificação das peças é realizada por sensores tipo indutivo e capacitivo.

A figura 3.1 apresenta uma foto do protótipo desenvolvido. Destaca-se que a construção do braço mecânico, circuitos de acionamento e sistema de controle, foi totalmente implementado a partir de materiais de baixo custo e sucatas.

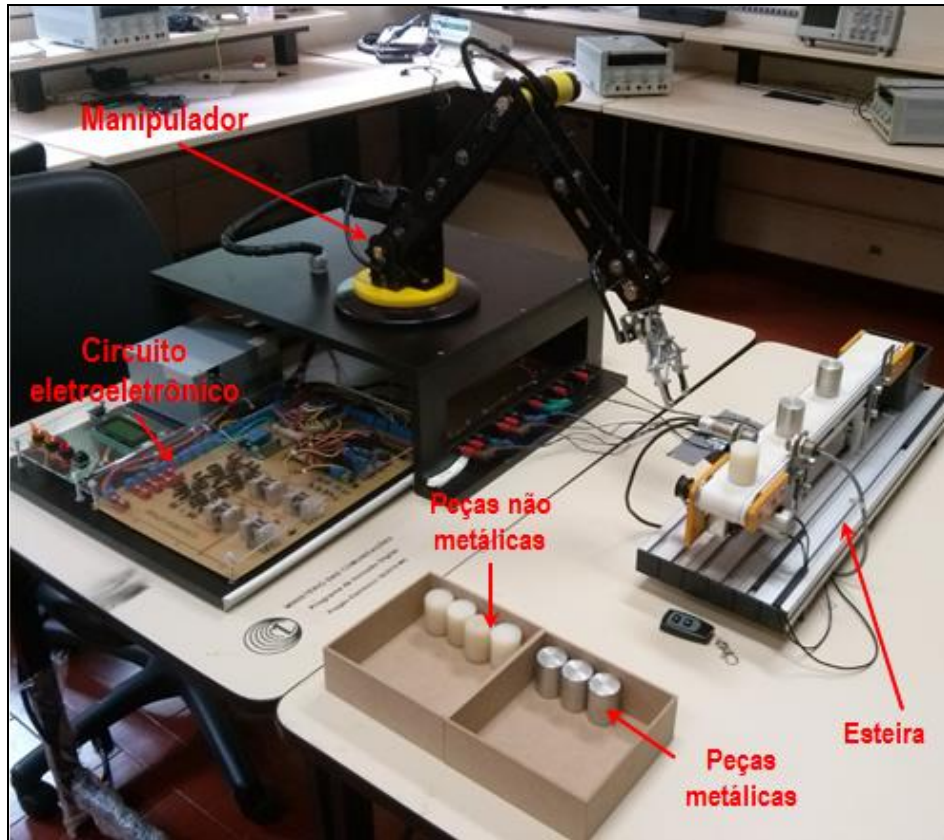


Figura 3.1 - Visualização do projeto.

A Figura 3.2 mostra o aplicativo desenvolvido especificamente para este trabalho, permitindo o controle e monitoramento em tempo real de todos os movimentos do braço e contagem das peças separadas.

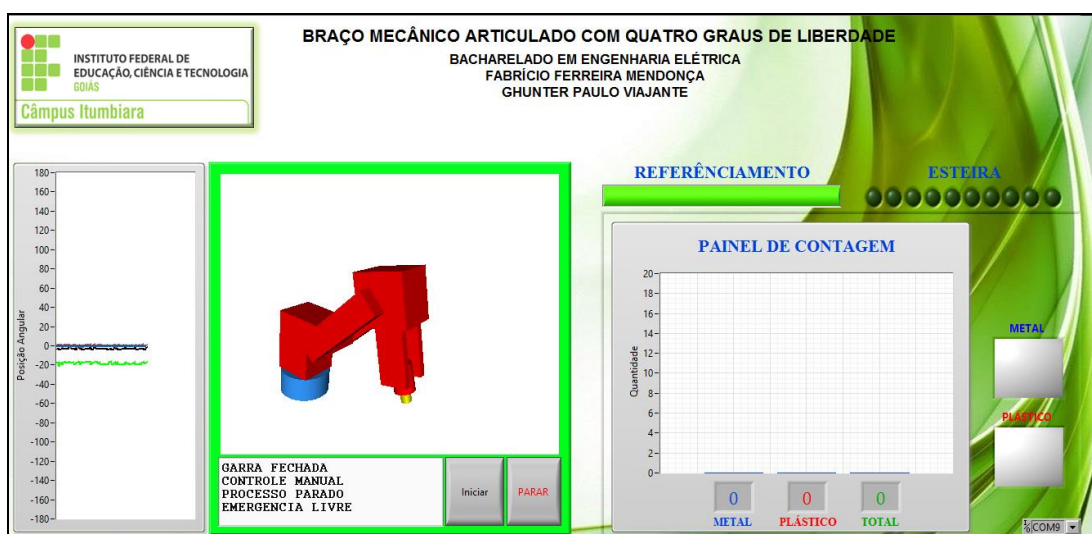


Figura 3.2 - Supervisório desenvolvido.

A figura 3.3 exibe a forma como o Arduino se relaciona com os demais dispositivos atuando como elemento controlador do projeto.

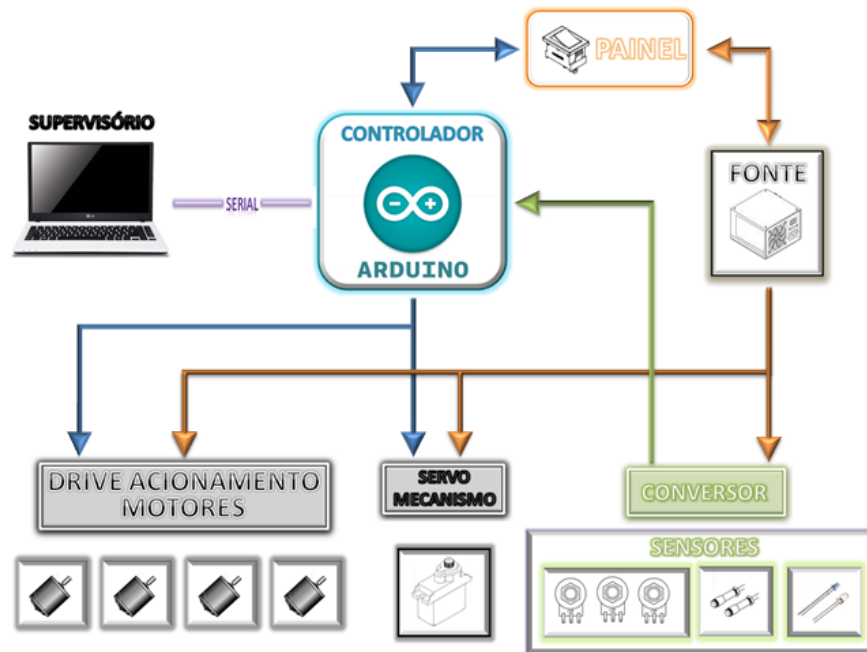


Figura 3.3 - Diagrama esquemático do projeto.

Todas as informações do processo são enviadas para sua CPU em forma de sinais elétricos analógicos e/ou digitais. O Arduino, por sua vez, recebe os sinais, processa e realiza as funções nas quais o microcontrolador está programado para executar. O Arduino dispara os drives de acionamentos dos motores enquanto recebe informações dos sensores implantados no manipulador e na esteira, assim como recebe também sinais elétricos do painel de comando para eventuais interrupções. O sistema supervisório opera junto com a plataforma estando sempre em comunicação com o controlador através da porta serial.

Em sistemas mecatrônicos são utilizados geralmente resolvers ou encoders absolutos para captura dos movimentos realizados pelo servomecanismo. Para suprir a demanda de um sensor de posição optou-se pela utilização de potenciômetros acoplados ao eixo de cada motor do manipulador devido ao seu baixo valor aquisitivo. Este ficou responsável por realimentar a malha de controle de posição de cada articulação.

Dentro do escopo também optou-se pela utilização de dois sensores externos para identificar o tipo de material que o braço robótico iria manipular. Para esta opção foi instalado um sensor de proximidade indutivo e capacitivo para identificar peças metálicas e não metálicas, respectivamente. Os sensores foram instalados em uma esteira rolante responsável por transportar as peças para o manipulador robótico.

Toda a parte de comando do braço ficou situada no painel de acionamentos para centralizar as informações solicitadas ao braço robótico. Este painel é composto por três chaves seletoras fixas de duas posições para habilitar o circuito, selecionar o tipo de operação desejado e bloquear todo o circuito em caso de falha, um botão de retorno por mola para referenciamento e navegação no display de LCD, um display de cristal líquido para exibir as informações do projeto e uma chave liga/desliga que alimenta a fonte de todo o circuito. A fonte foi recolhida através de materiais de informática, pois a mesma se tratava de uma fonte de computador *desktop* de 240W que não era mais utilizada.

A tarefa de controlar o braço robótico ficou para o Arduino DUE, um microprocessador de alta velocidade que possui 52 portas de entradas e/ou saídas digitais e de 12 portas de entradas e/ou saídas analógicas. A sua IDE é fácil de programar e aceita a Linguagem C como código fonte. Esse código contém todos os recursos necessários para funcionar o projeto, pois é ele que lê as informações que vem dos sensores e realiza as tomadas de decisão para cada aplicação.

3.1 ETAPAS PARA CONSTRUÇÃO DO BRAÇO ROBÓTICO

Para o desenvolvimento da estrutura mecânica do projeto foram adotadas as seguintes etapas:

- Escolha do motor a ser utilizada em cada junta;
- Desenvolvimento estrutural para definição do melhor material a ser utilizado;
- Construção e montagem das juntas e vínculos;

3.1.1 Escolha do Motor

Foi necessário para o projeto cinco motores para compor o acionamento mecânico das partes móveis do braço robótico com as seguintes aplicações:

- 01 motor para acionamento do disco da BASE;
 - 01 motor para movimentação da primeira articulação denominada OMBRO;
 - 01 motor para acionamento da segunda articulação denominada COTOVELO;
 - 01 motor para movimentação da terceira articulação denominada PUNHO;
-

- 01 motor para movimentação das engrenagens e articulações compostas para abrir e fechar a garra.

Optou-se pela utilização de motores de corrente contínua com redutores de velocidade acoplados, pois com a redução de velocidade é possível obter um controle mais preciso de rotação na ponta do eixo do motoredutor além de possuir um torque mais elevado do que um motor sem nenhuma caixa de engrenagem. Uma das vantagens de se utilizar um motor CC é a facilidade de manipular a tensão injetada em seus terminais. O controle consiste em variar o nível de tensão aplicado no circuito de armadura, e para tanto, foi apenas necessário elaborar um pequeno sistema eletrônico de acionamento CC com uma modulação PWM para controle de tensão.

Focando nos dispositivos de baixo custo, foi encontrado nos barracões e ferros velhos da cidade motores de acionamento de vidro elétrico de carro e motores para acionamento dos limpadores de para-brisas, pois ambos trabalham sob regime de tensão 12VCC e possuem um valor de torque elevado (cerca de 10N.m.). Para o OMBRO e para a o disco da BASE do projeto, foi escolhido o motor de vidro elétrico, figura 3.4, e o motor de limpador de para-brisa, figura 3.5, respectivamente, devido essas duas articulações serem as mais problemáticas por causa do alto torque exigido para movimentação e deslocamentos do braço robótico.



Figura 3.4 - Motor de vidro elétrico utilizado no OMBRO.

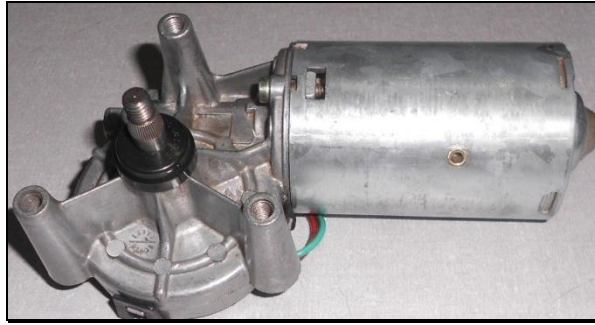


Figura 3.5 - Motor limpador de para-brisa utilizado na BASE.

A terceira e a quarta articulação foram utilizados motores de torque, tamanho e velocidades inferiores devido a baixa necessidade de força que o COTOVELO e o PUNHO exigiriam para realizar as movimentações necessárias. São representadas através das figuras 3.6 e 3.7.



Figura 3.6 - Motor DC utilizado no COTOVELO.



Figura 3.7 - Motor DC utilizado no PUNHO.

Para o acionamento da garra robótica foi utilizado um servomotor devido a facilidade de controle de posição. O servomotor escolhido é alimentado com tensões de 5V e recebe um sinal no formato PWM (*Pulse Width Modulation*) para fins de deslocamento do eixo. O circuito de controle do servo fica monitorando este sinal em intervalos de 20ms. Se neste intervalo de

tempo, o controle detecta uma alteração do sinal na largura do sinal, como ilustrado na figura 3.8, ele altera a posição do eixo para que a sua posição coincida com o sinal recebido.

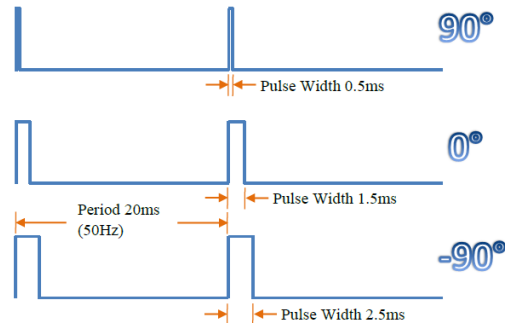


Figura 3.8 - Modulação de controle PWM para servomotores de baixa potência [16].

O servomotor já possui incorporado um decodificador de sinais para depurar a largura de pulso emitida pelo controlador e posicionar o eixo do motor na posição especificada. O sinal equivalente a posição do eixo já é conectada ao decodificador, fazendo o controle de posição internamente. Nesta configuração, a posição especificada pelo dispositivo de controle deve apenas trabalhar na largura de pulsos enviados para o dispositivo e o próprio servomecanismo se posiciona na posição adequada. A figura 3.9 ilustra o circuito integrado interno do servomotor utilizado.



Figura 3.9 - Servomotor utilizado na garra [16].

3.1.2 Estrutura Mecânica do Manipulador – Vínculos

Foi necessário escolher uma estrutura leve e ao mesmo resistente para suportar as forças atuantes no material que ocuparia a função de esqueleto, Link ou Vínculos, conforme algumas literaturas [6]. O objetivo estava focado na confecção de um projeto que não exigisse um alto valor de investimento e pudesse ser construído de forma fácil e barata. Para isso, foram utilizadas primeiramente chapas metal individuais que foram cortadas em formas de barras para compor a forma desejada do braço robótico que está vinculada a uma BASE rotativa, um Link de sustentação inferior interligando a o OMBRO com o COTOVELO e um Link superior conectando o COTOVELO com o PUNHO. A figura 3.10 ilustra a primeira tentativa de confecção do esqueleto.

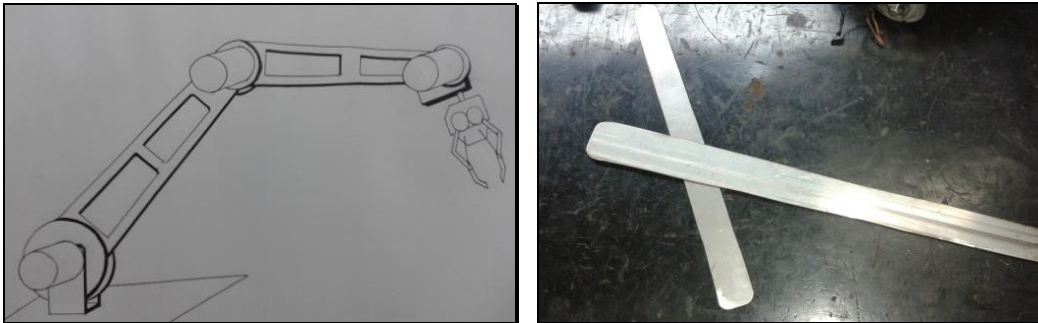


Figura 3.10 - Primeira opção de Link de sustentação.

Após cortado e montado o conjunto de maneira improvisada notou-se uma grande desestabilidade estrutural quando o conjunto realizava movimentos circulares, havia um alto índice de flambagem da peça no qual gerava um desconforto e uma descaracterização do projeto, além de existir uma grande dificuldade de controle de posição na ponta do braço robótico. Devido a isso essa opção foi descartada.

Em busca de um novo modelo para desenvolver o projeto foi encontrado, juntamente com materiais de sucata, pedaços de eletrocalhas e chapas perfuradas de zinco, naturalmente utilizadas em montagens industriais com objetivo de comportar cabos de energia elétrica para distribuição e ramificação de circuitos. Esse material se encaixou perfeitamente nas características necessárias para ser utilizado como estrutura, pois se tratava de um material leve devido as furações, flexível graças as paredes finas de 2mm e muito resistente a tração. Com a facilidade de manuseio dessa chapa foi escolhido uma nova estrutura mais diversificada do que a utilização de chapas únicas para o Link. A melhor composição foi um arranjo duplo para cada Link, sendo travadas com parafusos de 4mm de diâmetro transpassados nos furos decorrentes da característica da chapa de zinco. Quando realizado o intertravamento, a estrutura foi montada de

maneira improvisada novamente para verificar a rigidez do conjunto e seu comportamento durante os movimentos. O resultado obtido foi muito satisfatório, pois o novo formato mostrou uma grande estabilidade dinâmica. A figura 3.11 demonstra o modelo da nova estrutura definida.



Figura 3.11 - Estrutura final definida para seguir com o projeto.

3.1.3 Construção e Montagem do Manipulador Robótico

A confecção da estrutura foi constituída com as chapas de zínco perfuradas utilizadas em montagens de eletrocalhas. Após cortado, dobrado e fixado o corpo do link, foi necessário iniciar o processo de confecção das articulações que seriam distribuídas em quatro partes diferentes, coincidentemente compor os quatro graus de liberdade que o projeto possui.

A primeira junta confeccionada foi a BASE (ou cintura) onde, foi utilizando o motor de limpador de para-brisa e estendido o eixo de rotação, adaptado com solda elétrica com eletrodo revestido, uma ponta de parafuso 4mm com rosca para fixar o mesmo no disco giratório. A BASE é composta basicamente por três peças específicas: motor, disco giratório e base fixa de sustentação. A base fixa é um material circular de plástico encontrado no lixo para descarte, onde foram retiradas as sujidades e impurezas e adaptado para a deferida finalidade. O disco giratório foi confeccionado de material teflon disponível para uso em oficinas mecânicas e torneadoras de peças. Um torno mecânico foi utilizado para retificar compor a forma um ideal e de melhor

encaixe entre o disco giratório e a base de fixação. Após realizado o encaixe entre as duas peças, foi instalado o motor elétrico para dar movimento ao disco que sustentará todo o braço robótico sendo este o primeiro grau de liberdade adquirido para o protótipo. A figura 3.12 ilustra o procedimento descrito.



Figura 3.12 - Conjunto da BASE.

Para compor a articulação do OMBRO foi necessário cortar e dobrar uma chapa de aço carbono com o objetivo de sustentar o motor de vidro elétrico de carro e os pinos de travamento da articulação de tração. Foi necessário confeccionar um eixo extensivo para adaptar no motor de acionamento desta articulação. Este eixo é feito de tecnil e também foi necessário realizar usinagem para moldar a peça de maneira com que encaixasse no motor e conseguisse sustentar toda a estrutura do OMBRO, apoiando de um lado o motor e do outro o mancal de sustentação. Em seguida, imagem 3.13 ilustra a imagem dos acoplamentos usinados.



Figura 3.13 - Articulação OMBRO.

Para a confecção do COTOVELO e do PUNHO também foi necessário fazer uso do torno mecânico para confeccionar mais eixos extensivos de tecnil, onde, de um lado acopla-se o motor DC e do outro lado a segunda chapa de sustentação. Para todas as articulações montadas foi preciso instalar um potenciômetro com o objetivo de captar a posição angular do eixo no momento de rotação e servir também como um segundo mancal de apoio para o eixo extensivo de tecnil. As figuras 3.14 e 3.15 mostram as articulações descritas. A garra e o servomotor foram incorporados ao final da montagem, sendo instalado na juntamente ao PUNHO, visto também na figura 3.15. O manipulador completo foi concluído sendo composto então por quatro motores de acionamento DC para movimentação das juntas e um servomotor para abertura e fechamento da garra.



Figura 3.14 - Articulação COTOVELO.



Figura 3.15 - Articulação PUNHO.

Para fins de acabamento o projeto foi submetido a pintura via spray e colocado sobre uma base de madeira para acomodar o braço mecânico na parte superior e a placa de circuito

impresso na parte inferior com o auxílio de roldanas para embutir e acondicionar a placa de circuito impresso. Abaixo é informado a sequência no qual o projeto foi montado:

- Pintura das partes individuais;
- Montagem da BASE;
- Montagem das Juntas;
- Passagem dos cabos de alimentação dos cabos de força e comunicação;

A figura 3.16 demonstra as etapas citadas acima e a figura 3.17 ilustra um desenho desenvolvido em plataforma AutoCAD.



Figura 3.16 - Acabamento Final.

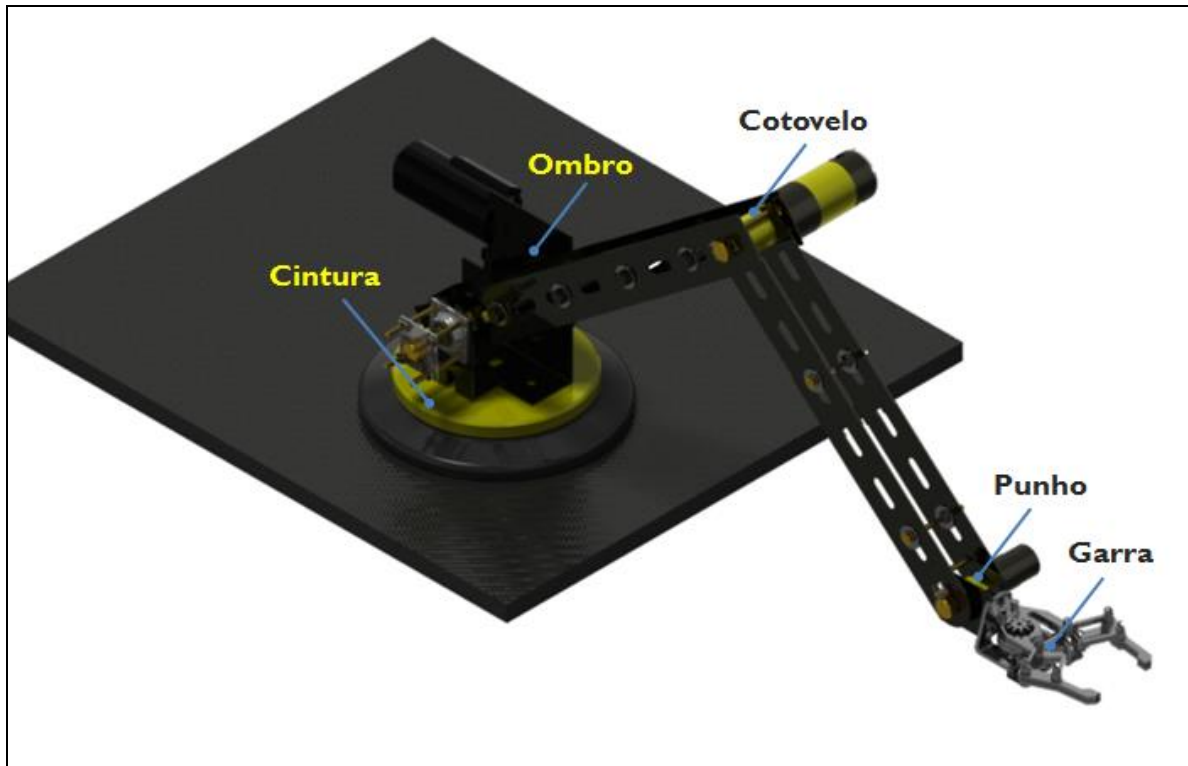


Figura 3.17 - Desenho do manipulador em plataforma AutoCAD.

3.2 ESTRUTURA ELETROELETRÔNICA

Todos os motores adotados para o projeto necessitam de uma tensão de 12V em corrente contínua, com exceção do servomotor da garra que trabalha com 5V. A alimentação do Arduino pode ser direcionada através da porta de alimentação de dados USB ou por uma fonte externa de 9-12V e suas portas digitais e analógicas operam com uma tensão de 3,3V. O display de cristal líquido, que exibe as informações para referenciamento, exige uma alimentação de 5V. Pode-se notar que existe uma grande variedade de níveis de tensão distribuídas sobre todo o sistema de alimentação do projeto. Para suprir todos os níveis de tensão exigidos pelo circuito do manipulador, foi adotado uma fonte de computador com saídas de 12, 5 e 3,3V. Neste levantamento, ainda podemos incluir o motor da esteira externa de alimentação de peças, o motor da esteira de saída, os sensores indutivos e capacitivos, os fotodiodos que estão recolhendo sinal da BASE e os LED's de iluminação da planta.

A corrente nominal dos motores da BASE e do OMBRO operam em uma faixa de 5A, porém, em momentos de muito esforço físico, como subida dos vínculos por exemplo, a corrente pode atingir picos maiores do que 10A. Para acionar esses motores se faz necessário construir

drives de acionamentos para que o microcontrolador consiga disparar os transistores e o sistema consiga atingir esses níveis de potência.

Os sensores indutivos e capacitivos instalados na esteira, trabalham sob um regime de operação de 9-30V e emitem um sinal de nível alto ou baixo equivalente ao valor de alimentação de tensão, logo, para que estes sensores possam enviar sinais para o Arduino sem danificá-lo, foi necessário construir um condicionador de sinal com o objetivo de reduzir o nível de tensão de 12V para 3,3V. O mesmo equivale para os fotodiodos que foram inseridos na BASE para controle de posição da cintura.

3.2.1 Drivers de Acionamento dos Motores – Ponte H

Os motores de maior potência são os motores da BASE (Cintura) e do OMBRO devido as suas localizações que exigem maior esforço físico em regime de operação. O motor da BASE deve ter potência suficiente para girar a cintura do manipulador que por sua vez carrega todo o peso do equipamento. Já o motor elétrico do OMBRO deve suportar o peso de toda a estrutura e ter rigidez para suportar as forças e vibrações que aparecem quando as outras articulações estão se movimentando. Devido a sua potência e sua robustez, esses dois motores são os maiores consumidores de energia do projeto e solicitam 10A de corrente em movimentos que exigem muito esforço.

Para estes dois casos foram escolhidos os transistores TIP142 (NPN) e o TIP147 (PNP) para compor o circuito ponte H juntamente com o BC337 que auxilia na amplificação do sinal para saturar os TIP's. A escolha destes transistores foi baseada na capacidade de condução de corrente, pois estes dois dispositivos (TIP142 e TIP147) trabalham com 10A de corrente nominal suportando 15A em situações de pico. A ponte H exige dois comandos elétricos para acionar as chaves posicionadas em diagonal para fechar o circuito e alimentar o motor elétrico com energia do barramento CC. A figura 3.18 demonstra o tipo de configuração adotada para acionar os motores.

O sinal 3,3V do microcontrolador satura o transistor BC337 que por sua vez polariza os TIP's 147 e 142 permitindo a saturação do circuito de potência e acionando o motor com 12V em condições nominais. Os transistores foram escolhidos para chavear o circuito devido à praticidade de se trabalhar utilizando o sistema de chaveamento baseado no controle de largura

de pulsos PWM (*Pulse Width Modulation*). Além de existir a possibilidade de se trabalhar com diferentes sentidos de rotação do motor ainda é possível alterar a sua velocidade.

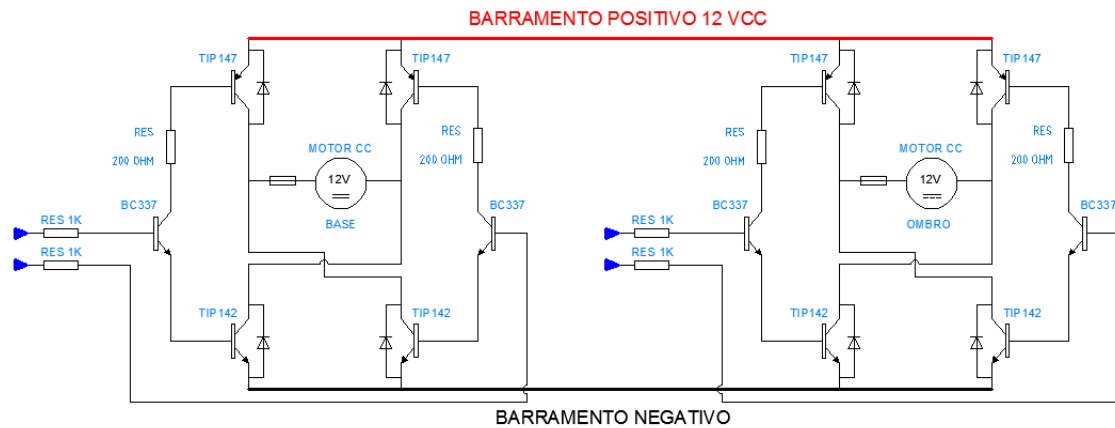


Figura 3.18 - Drive acionamento BASE / OMBRO.

A mesma configuração foi adotada para acionar os motores do COTOVELO e do OMBRO, porém para estes dois motores foram utilizados transistores de menor capacidade de condução devido a baixa potência consumida pelos motores. O circuito e a configuração são idênticos quando comparados com o circuito do driver de acionamento dos motores da BASE e do OMBRO, só que para esta finalidade, o circuito eletrônico foi projetado utilizando os transistores TIP122 e TIP127, que trabalham com 5A de corrente nominal.

O acionamento do servomotor é feito através de sinal PWM, sem existir a necessidade criação de drivers, pois este circuito está disponível dentro do servomotor, dispensando a elaboração do mesmo. Sendo assim, para o circuito elétrico de acionamento do servomotor foi necessário apenas alimentar o dispositivo com 5V e um sinal PWM do controlador para controle de posição. A figura 3.19 ilustra esta configuração.

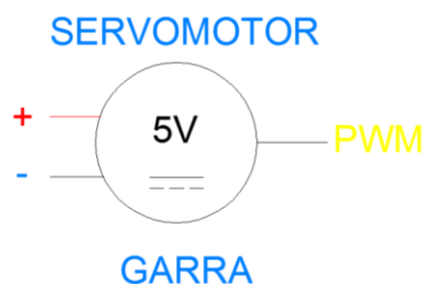


Figura 3.19 - Acionamento servomotor.

3.2.2 Circuito Elétrico do projeto

Além do driver de acionamento dos motores elétricos foi necessário realizar diversas outras configurações para trabalhar com as tensões de saída do Arduino DUE. Os sensores de proximidade (Indutivo e Capacitivo) foram ligados na mesma fonte de alimentação dos motores elétricos, em um barramento de 12V, logo, o sinal de saída do sensor que é recebido pelo Arduino teve que ser trabalhado de forma diferente, pois, como já mencionado, os pinos de entrada da placa do microcontrolador não suporta esse nível de tensão. Para fazer os sensores se comunicarem com o Arduino foi preciso projetar um circuito divisor de tensão que regulasse um sinal de 12 para 3,3V. O circuito equivalente montado está representado no diagrama elétrico da figura 3.20.

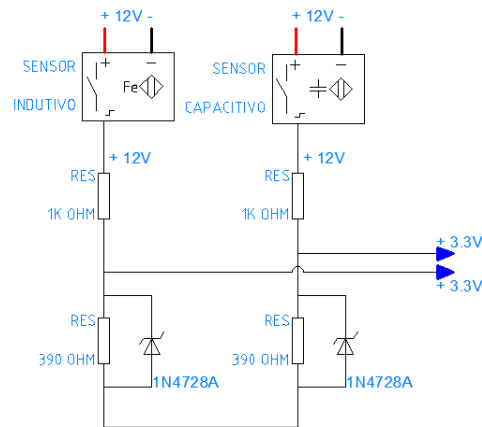


Figura 3.20 - Circuito divisor de tensão.

Além dos resistores foi adicionado um diodo Zenner 1N4728A para estabilizar o nível de tensão em 3,3V garantindo a qualidade do sinal enviado ao microcontrolador.

A cintura do braço robótico acabou se tornando um local de difícil manuseio impossibilitando adicionar algum potenciômetro para captura de movimento rotacional. Neste caso foi adotada uma estratégia baseada no princípio de funcionamento dos encoders industriais incrementais. Foi inserido um diodo emissor de luz infravermelho em baixo do disco da cintura com manchas escuras para que o feixe de luz seja interrompido e outrora tenha a passagem liberada para emissão de luz. Um segundo diodo foi acrescentado para que fosse capaz de receber estes sinais emitidos pelo diodo emissor de luz infravermelho. A configuração mencionada está representada na figura 3.21. Os diodos utilizados são o TIL 32 trabalhando como emissor e TIL78 trabalhando como receptor da luz.

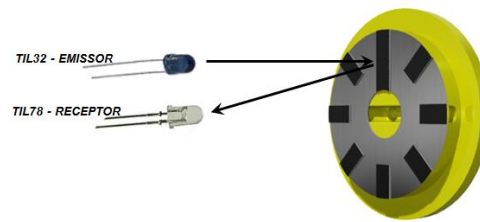


Figura 3.21 - Adaptação dos sensores para a BASE.

Os diodos estão fixos internamente na base em baixo do disco giratório e quando o motor gira a BASE e movimenta todo o braço robótico o disco acompanha o movimento e interrompe o sinal de feixe luz que antes estava sendo enviado para o receptor. Os pulsos gerados pela movimentação do disco são enviados para o microcontrolador e serve como referencia de posição da BASE. O desenho em corte da figura 3.22 ilustra a posição dos sensores.

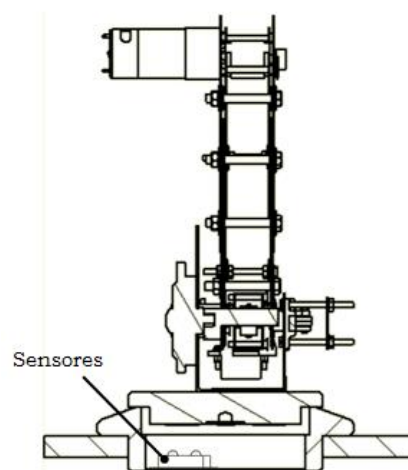


Figura 3.22 - Vista em corte da posição dos sensores.

Os diodos de luz infravermelho funcionam melhor com uma tensão de 5V do que quando utilizado 3,3V, logo, para utilizar o sinal como referencia ao Arduino foi preciso adaptar um circuito divisor de tensão em série com o sinal de saída equivalente do circuito TIL78. Este circuito é ilustrado na figura 3.23.

O acionamento dos motores das esteiras de entrada e saída de peças são acionados via transistor como segue o diagrama da figura 3.24.

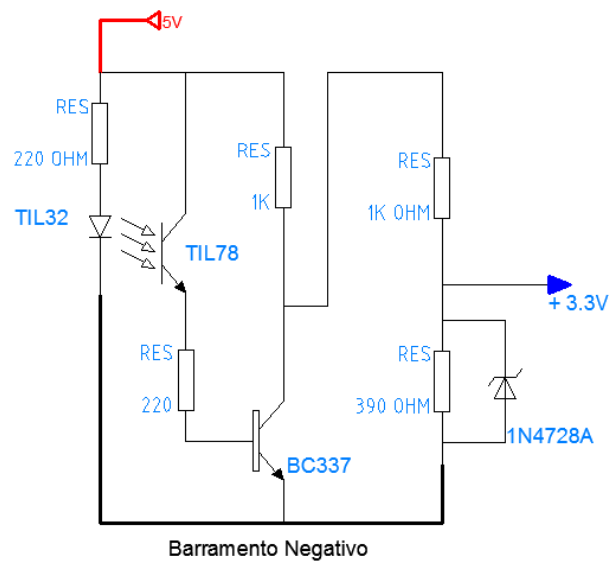


Figura 3.23 - Circuito TIL32 e TIL78.

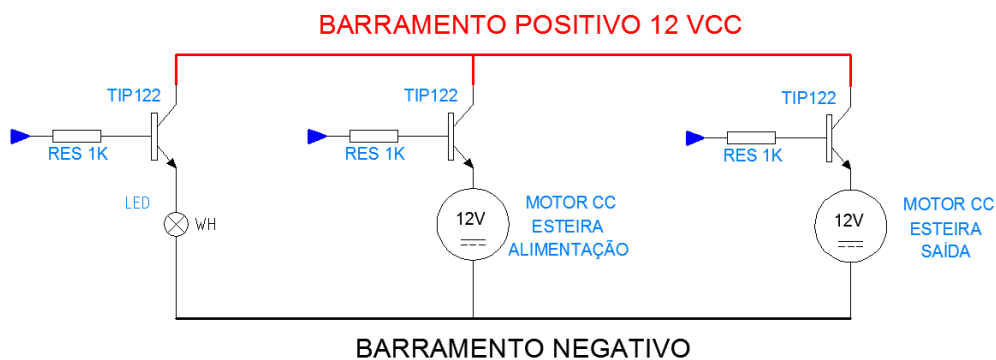


Figura 3.24 - Acionamento dos dispositivos externos.

Para as chaves de comando instaladas no painel adotou-se o sistema *Pull Up* para acionar as portas digitais do microcontrolador. Dessa maneira o dispositivo de controle armazena a informação caso alguma porta em específico seja acionada. No mesmo barramento de 3,3V estão conectados os potenciômetros de controle de posição dos motores elétricos. Cada potenciômetro está ligado em uma porta específica de leitura analógica para análise de sinal contínuo de tensão. A figura 3.25 mostra o esquema elétrico para as chaves de comando.

A fonte de computador adotada para este projeto possui várias saídas para diferentes níveis de tensão. No início do projeto, quando o movimento das articulações estavam em fase de teste elétrico, foi utilizado um terminal com fornecimento de energia a nível 3,3V da fonte de computador, porém em circunstâncias de grande esforço elétrico dos motores da BASE e do OMBRO, ocasionava uma pequena queda de tensão nos terminais de 3,3 e 5V. Como os

potenciômetros estavam conectados neste barramento (3,3V) o sinal de referencia de posição sofria distúrbios e informava valores incorretos. Para não ocorrer este tipo de problema, a fonte de 3,3 e 5V foi retirada do próprio Arduino DUE, que possui três terminais disponibilizando 5V, cinco terminais para terra (GND) e um terminal fornecendo 3,3V.

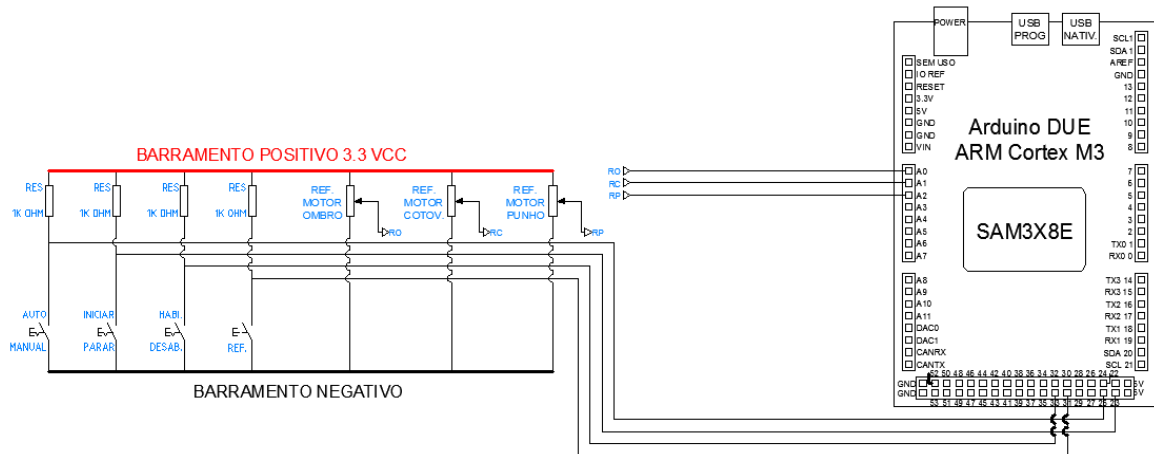


Figura 3.25 - Circuito Pull Up e analógicas.

O OMBRO exige da fonte grande disponibilidade de potência para realizar seus movimentos devido ao peso do conjunto formado pelo COTOVELO e o PUNHO. O potenciômetro responsável por fazer essa leitura está conectado na porta A0 do microcontrolador. Quando o OMBRO realizava os movimentos de subida e descida exigia-se uma grande quantidade de potência a ser fornecida pela fonte ocasionando uma queda de tensão.

A figura 3.26 ilustra o distúrbio sofrido pelo sinal analógico durante a queda de tensão, onde o sinal azul é o valor que deveria ser lido e o sinal vermelho é o valor real na porta analógica.

A figura 3.27 ilustra o valor da posição do motor do OMBRO quando a alimentação do potenciômetro é transferida para o Arduino inibindo as interferências causadas pela fonte tornando o controle praticamente estável, sendo possível realizar os movimentos de forma correta.

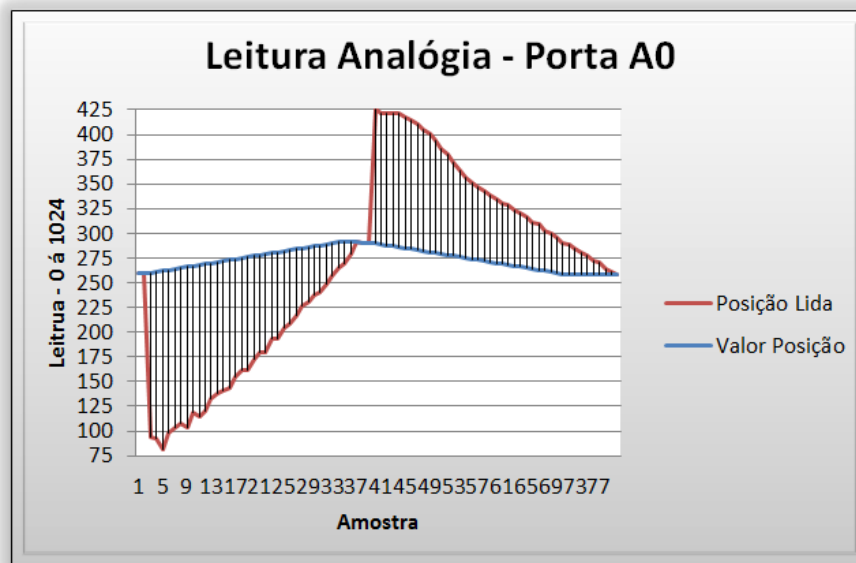


Figura 3.26 - Sinal fornecido pelo Potenciômetro com tensão de referencia fornecida pela fonte.

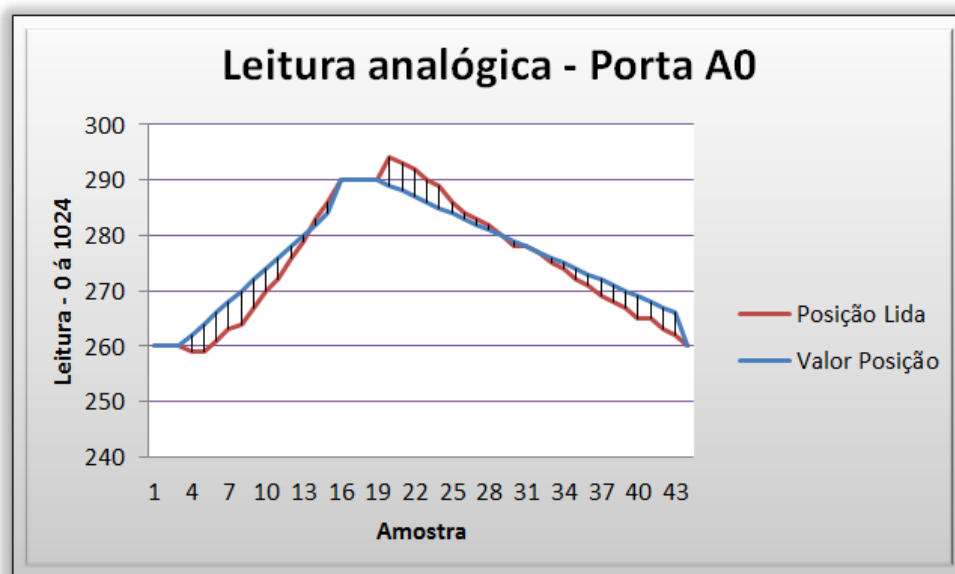


Figura 3.27 - Sinal fornecido pelo Potenciômetro com tensão de referência fornecido pelo Arduino.

3.2.3 Placa Eletrônica de Acionamento do Manipulador

O circuito impresso é o meio mais comum e prático usado para montagens definitivas de circuitos eletrônicos. Os processos de confecção de placas de circuito impressos são vários, podendo distinguir-se basicamente em processos caseiros (acessíveis a qualquer pessoa) e

processos profissionais (mais elaborados, que possibilitam uma produção industrial em larga escala). O termo circuito impresso é derivado do método original, onde um padrão gravado ou estampado é usado para mascarar as áreas desejadas, de uma placa de cobre.

Para o desenvolvimento da placa eletrônica foi utilizando o software Isis Proteus que permite a montagem e simulação de circuitos eletrônicos. Neste, é possível dimensionar componentes e recrutar os dispositivos conforme a necessidade do projetista. A vantagem do Isis Proteus, além da possibilidade de simular os circuitos, é conseguir desenhar e visualizar e esquema eletrônico, caso a placa seja confeccionada.

Uma vez concluído e simulado o circuito eletrônico é possível utilizar a extensão Ares Proteus para visualizar a disponibilidade das trilhas e a melhor forma de distribuir os dispositivos na placa, fisicamente falando. A figura 3.28 ilustra o circuito da ponte H utilizada neste projeto desenhada no Isis Proteus.

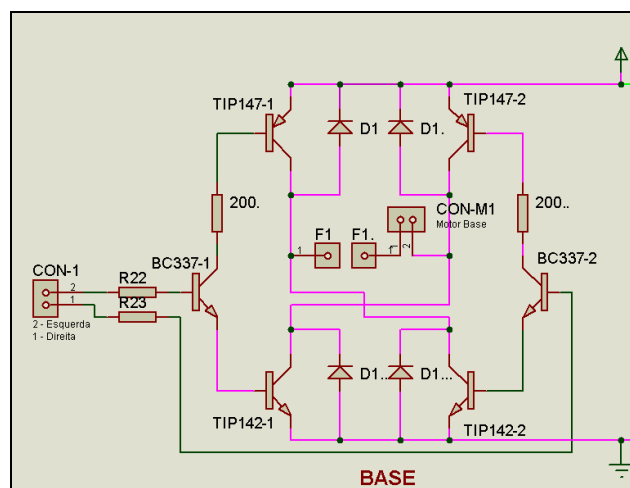


Figura 3.28 - Esquema da ponte H no ISIS Proteus.

A largura da trilha depende da ordem de grandeza das correntes envolvidas no circuito. O efeito causado pela corrente manifesta-se de duas formas: dissipação de potência na forma de calor e diferenças de potencial, ambas nocivas ao projeto da placa. O cálculo da largura da trilha pode ser realizado através da *resistência elétrica da trilha* considerando a largura, a espessura e o comprimento da trilha, pode ser feito através dos critérios adotados por *queda de tensão* e pode também ser dimensionada pelo *critério da densidade de corrente*. Para este projeto foi adotado o cálculo por densidade de corrente [19].

No caso de trilhas de circuito impresso, o valor máximo aceitável para a densidade de corrente (J) é de 35 A/mm², bem acima dos níveis permissíveis ao condutor cilíndricos comuns,

graças á própria geometria plana das trilhas, que proporcionam maior capacidade de dissipação de potência por unidade de área [19]. A equação da densidade de corrente utilizada é expressa pela equação 3.1.

$$J = \frac{l}{a \cdot b}$$
3.1

Onde: J = Densidade de corrente;

l = corrente elétrica em ampère;

a = espessura da trilha em mm;

b = largura da trilha;

Para uma densidade de corrente fixa em 35A/mm², uma espessura de 0,05mm, que é geralmente empregado nas placas de cobre, e uma corrente que pode atingir níveis próximos á 10A encontrou-se uma largura de **6mm** para o circuito correspondente ao motor da BASE e do OMBRO. A mesma metodologia foi empregada para dimensionar a trilha dos motores do COTOVELO, PUNHO e das esteiras transportadoras. Na figura 3.29 é possível ver a disposição dos dispositivos na placa e na imagem 3.30 o diagrama eletrônico do circuito completo.

Devido ao grande número de trilhas e as diferentes larguras, que deveriam ser respeitadas decorrente do dimensionado, a placa foi confeccionada por uma empresa que trabalha com plotagens e confecções de placas de circuito impresso. O resultado desse trabalho está expresso na figura 3.31.

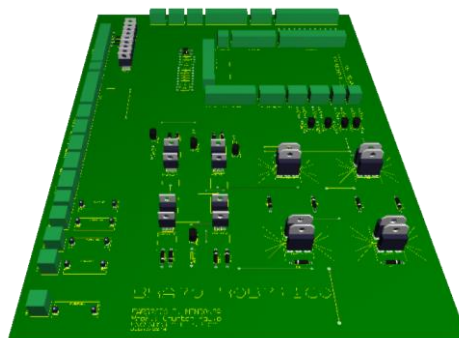


Figura 3.29 - Layout da placa disponibilizada pelo ARES Proteus

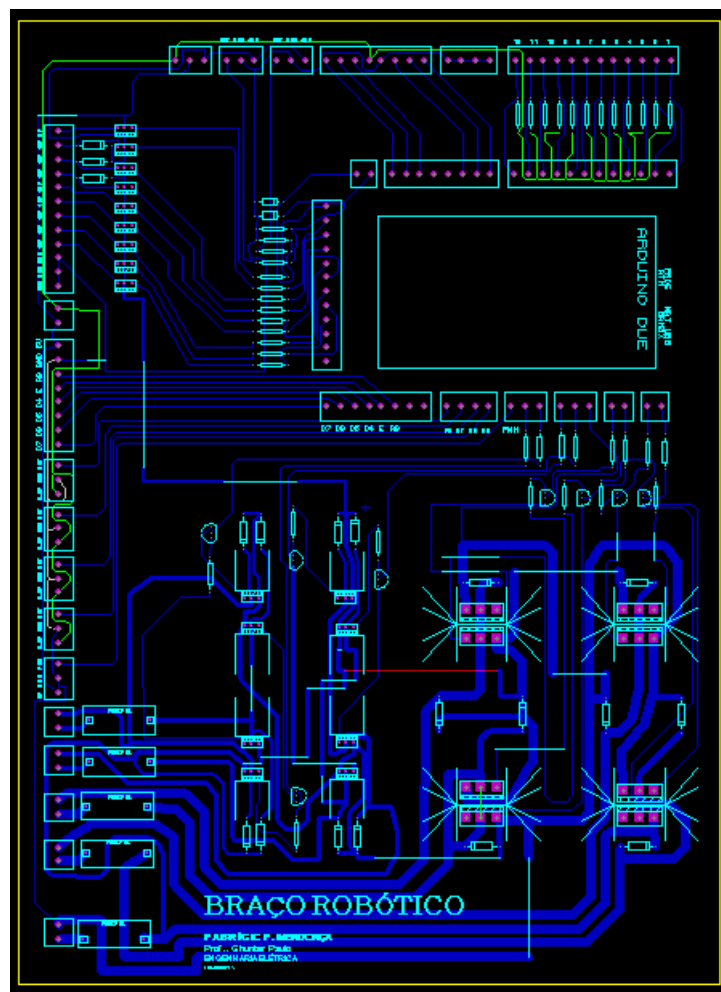


Figura 3.30 - Projeto para confecção da placa.

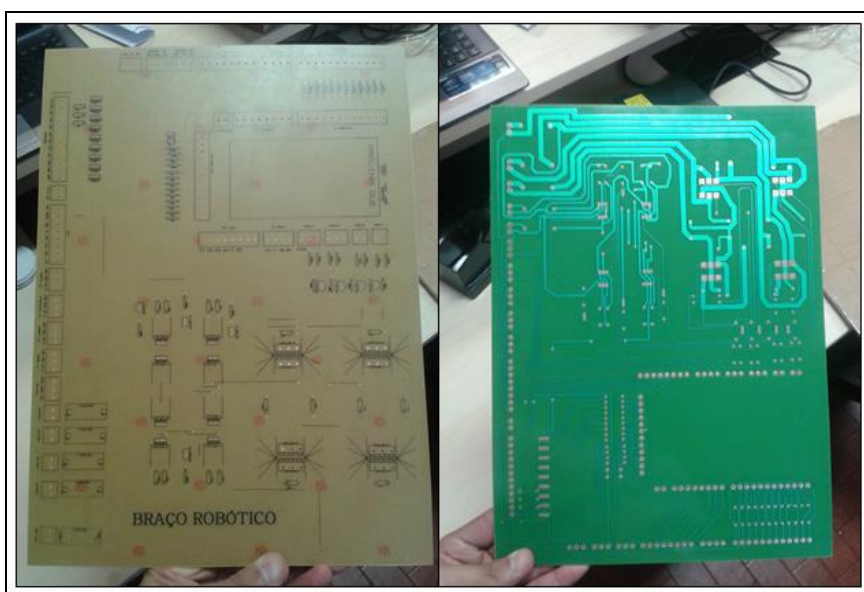


Figura 3.31 - Frente e verso da placa eletrônica pronta.

A figura 3.32 mostra o arranjo final do circuito eletroeletrônico do projeto.

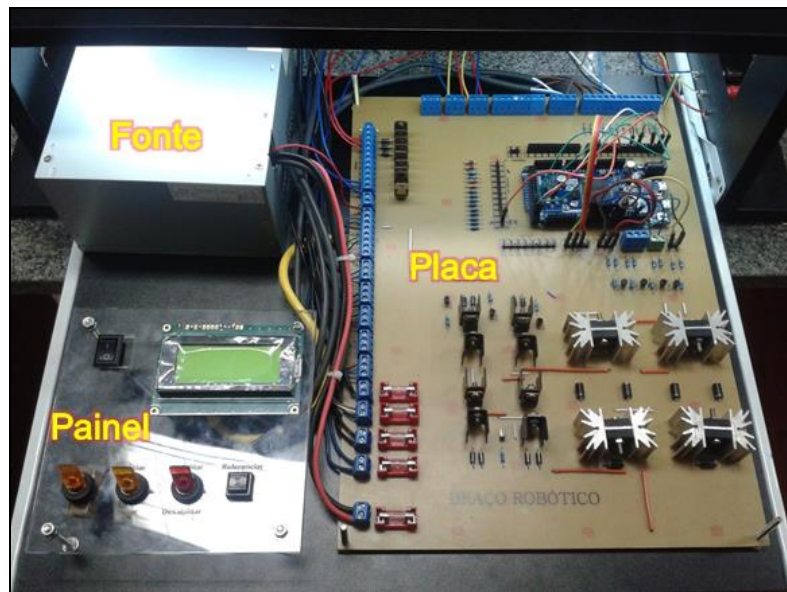


Figura 3.32 - Circuito Eletroeletrônico.

3.2.4 Programação e Controle Braço Robótico

A plataforma Arduino exige um software específico de programação onde, basta selecionar a porta USB que o microcontrolador está conectado e o tipo de board que o programador está usando. Como foi utilizado o Arduino DUE foi necessário utilizar a versão 1.5.8 da *IDE ARDUINO* devido este microcontrolador ser um ARM Cortex-M3 que somente versões mais atualizadas são capazes de comunicar com este dispositivo. O Arduino DUE possui duas portas para comunicação serial, a *Programming PORT* e a *Native PORT*. Para descarregar o programa foi necessário conectar apenas a porta de programação.

A grande vantagem da plataforma Arduino, além da vasta biblioteca de funções prontas, é a praticidade de descarga de dados na memória do microcontrolador. Todo *download* de dados pode ser feito apenas conectando o cabo USB e toda a transição é feita via comunicação serial sem necessitar de placas específicas de programação, assim como acontece com o PIC por exemplo.

A linguagem de programação utilizada foi a Linguagem C vinculada a alguns comandos restritos da IDE do Arduino devido às bibliotecas e a própria funcionalidade do programa. Para

programar o braço robótico foi necessário utilizar 5 portas analógicas e 27 portas digitais funcionando algumas como entrada de dados e outras como saída. Dentre as portas digitais utilizadas, 9 foram destinadas para acionar os motores com sinais PWM.

Conforme ilustrado no fluxograma da programação na figura 3.33, quando o microcontrolador é iniciado ele tem a opção de entrar em uma rotina de referenciamento que nada mais é que atualizar as portas analógicas com os valores lidos nos potenciômetros para que o programa consiga enxergar a real posição que o braço se encontra. Após o referenciamento existe uma função que aciona os drives dos motores e de acordo com o referenciamento ele se posiciona em uma posição de repouso, chamada de posição inicial, e nesse momento o código aciona a porta equivalente ao motor da esteira de alimentação de peças fica aguardando a chegada de algum material. Neste instante o programa fica dentro de um loop apenas fazendo varreduras das portas que recebem sinal do painel de comando e do sensor indutivo e capacitivo fixados na esteira para detecção de peças.

Caso alguma peça seja identificada, através do sinal que o sensor capacitivo enviou para o microcontrolador, a esteira é desligada e é feito uma verificação se existe algum bit de armazenamento indicando que o sensor indutivo também foi acionado. Caso o sensor indutivo mande um pulso indicando que por ali também passou uma peça de metal o código registra que aquela peça é uma peça Metálica. Caso o sensor capacitivo seja acionada mas nenhum bit foi registrado anteriormente informado que o material é de metal o código armazena a informação que aquela peça é Não metálica.

Uma vez identificado o tipo de material que está na esteira, o motor é desligado para que a peça detectada fique em repouso para que então o braço mecânico possa recolher a mesma. Existem funções diferentes para cada tipo de motor utilizado no projeto junto com suas respectivas velocidades. O microcontrolador aciona os drivers em uma sequência já programada referente a qual junta entrará em funcionamento primeiro para atingir a posição já determinada, que é a posição de recolher o material.

O programa já está configurado para seguir com a segunda posição onde, dependendo do tipo de material que foi identificado é estipulado o acionamento dos motores para transporte da peça. Através dos dispositivos de leitura analógica (potenciômetro) é possível identificar as posições em que o braço está localizado e a qualquer momento retorná-lo para a posição inicial. É isso que acontece quando o braço deposita a peça na caixa correspondente ao material, metal ou não metal.

Como o conversor A/D do microcontrolador possui 10 bits de resolução, o sinal analógico recebido do potenciômetro e lido nas portas analógicas trabalham na faixa entre 0 e 1024. Na fase de referenciamento o programa captura as posições iniciais do braço e armazena para utiliza-los quando for acionar algum motor.

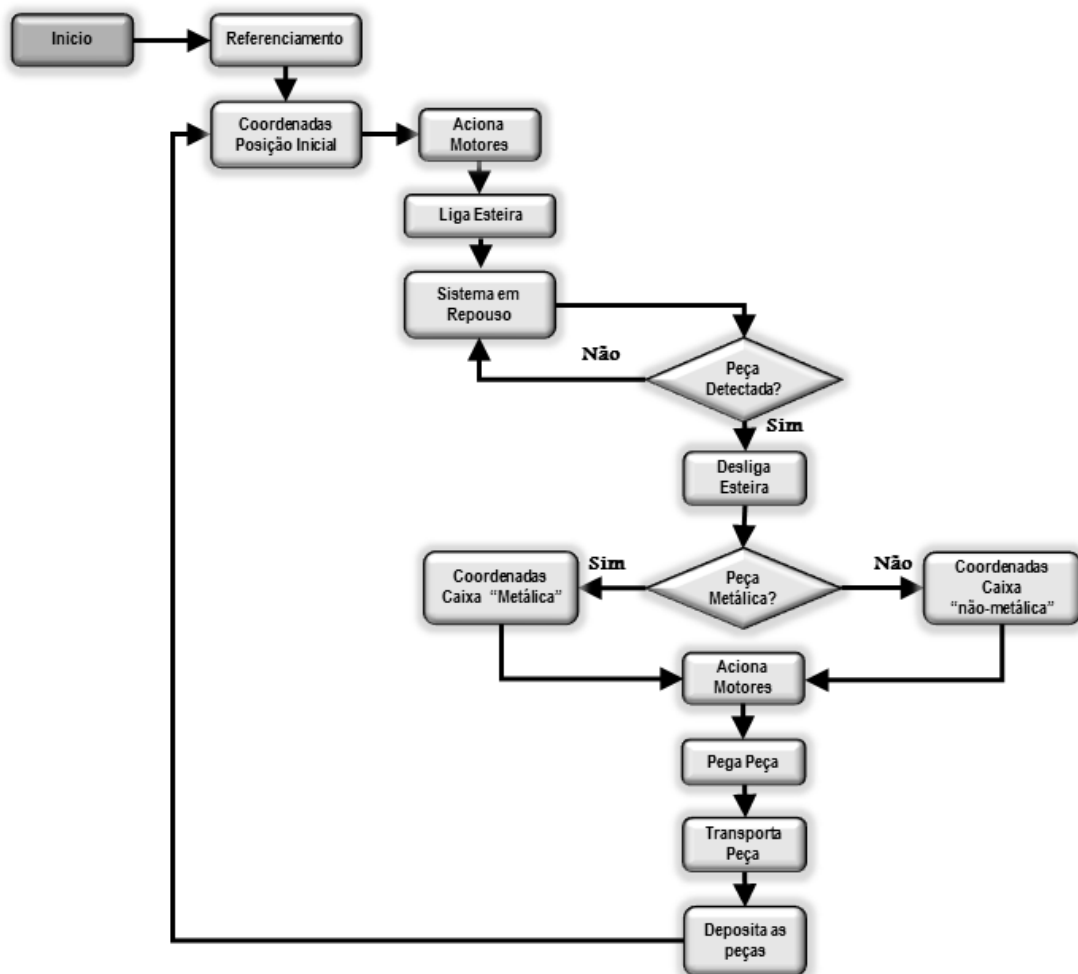


Figura 3.33 - Fluxograma da rotina programada.

Foi implementado para os motores um controle de posição Proporcional, isto é, o erro é acrescentado no compensador e de acordo com a distancia que o eixo do motor está em relação ao setpoint estipulado, é o valor com que o atuador vai atribuir para velocidade. Em circunstâncias normais, o movimento da junta se torna mais suave durante as atividades, pois quanto mais perto o eixo estiver do setpoint, menor será a velocidade atribuída e mais suave será

o final do movimento sem causar trancos ou vibrações excessivas. A equação 3.2 descreve a expressão de um compensador PID:

$$PID = K_p \cdot erro(t) + K_i \cdot \int_0^t erro(t) \cdot dt + k_d \cdot \frac{derro(t)}{dt} \quad 3.2$$

Para o projeto foi utilizado apenas o Proporcional e o erro foi a diferença entre o Setpoint e o valor lido quando algum motor era acionado. É possível visualizar no capítulo 4, “Resultados Experimentais”, o motivo pelo qual optou-se apenas pelo controle Proporcional.

O potenciômetro acoplado ao eixo central da junta é quem acompanha a rotação do motor e fornece um sinal de caráter analógico para que o programa consiga identificar a posição do eixo do motor. A malha representativa deste controle esta apresentado na figura 3.34.

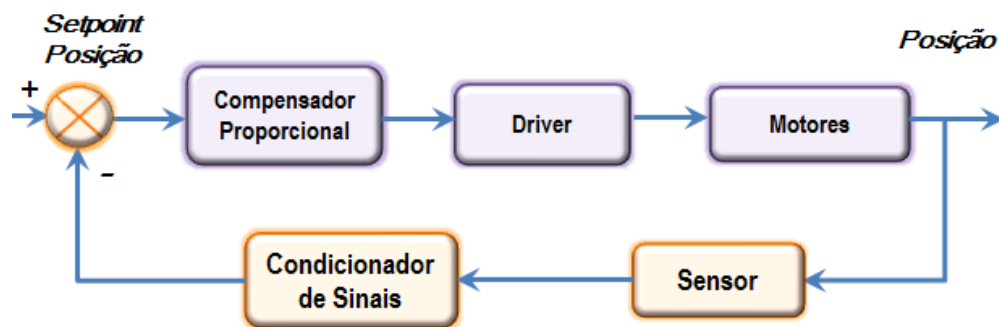


Figura 3.34 - Malha de controle de posição dos motores.

Quando o microcontrolador identifica uma necessidade de movimento um setpoint é estabelecido para a primeira articulação que entrará em movimento. A porta analógica correspondente a junta em movimento é constantemente lida para amostrar os valores referente a posição do mecanismo. O controlador calcula o erro do sistema, através da equação 3.3, para que o mesmo possa somar na equação da velocidade (3.4) e determinar o valor do sinal PWM que atuará nos drives de acionamento dos motores.

$$erro = Setpoint - Valor\ lido \quad 3.4$$

$$Velocidade = KP.erro + Vi$$

3.5

V_i é uma velocidade mínima estipulada como velocidade inicial para acionamento do motor. O controle de velocidade é agregado na rotina por dois motivos: Primeiro, atuar na suavidade do movimento. Segundo, atuar no controle de posição, pois quando a posição é atingida, a velocidade é igual a zero.

Para cada motor existe uma velocidade mínima de partida que deve ser respeitada porque dependendo da velocidade inicial adotada o motor não consegue vencer o momento de inercia devido ao peso realizado do vínculo. A própria dinâmica do sistema não permite tal manobra. A velocidade é convertida em um valor correspondente ao PWM em uma faixa que varia de 0 á 255 correspondente aos 8 bits de resolução nas portas digitais com saídas PWM. Na figura 3.35 é ilustrado o sinal de PWM, recolhido pelo osciloscópio, para acionamento do PUNHO para realizar um movimento com 78% da velocidade, equivalente a uma saída de 200/255 em uma frequência de saída de 1KHz.

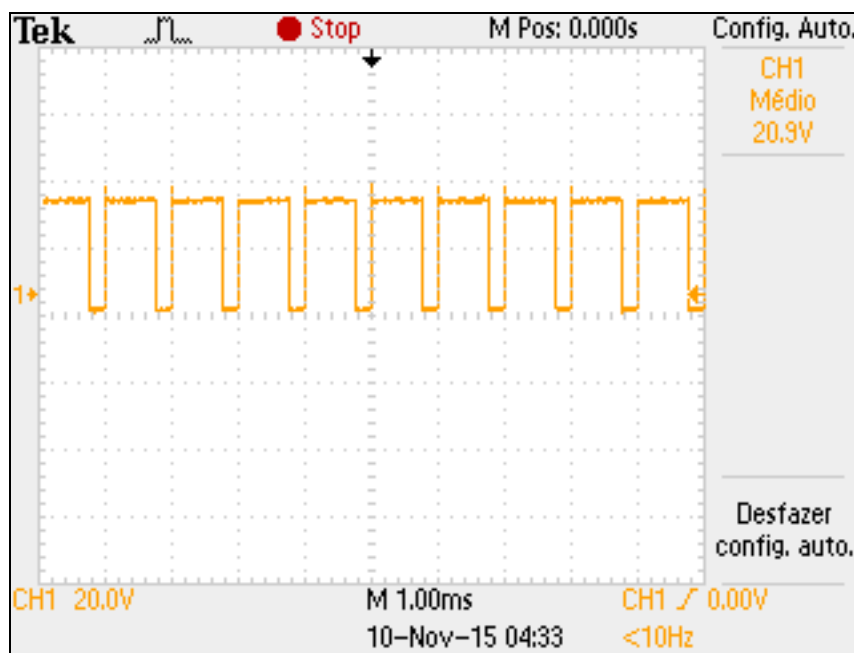


Figura 3.35 - Sinal PWM com 78% da velocidade.

O display de LCD instalado no painel de comando exibe os dados do projeto como ano, instituição de ensino e participantes diretos e indiretos que auxiliaram no desenvolvimento do projeto. As informações estão em um loop que só pode ser desfeito quando o botão de

referenciamento é apertado. Este botão pode ser acionado tanto no painel de comando quanto no botão do controle remoto que foi adaptado para realizar comandos a distância.

Dentro do código possui também uma rotina de comunicação serial para que o microcontrolador possa enviar dados para o computador com o objetivo de atualizar o supervisor que foi elaborado para acompanhamento das atividades que o robô está executando. Dentro do protocolo são enviadas as seguintes informações:

- Sinal informando qual posição a cintura se encontra;
- Valores analógicos correspondentes as posições do OMBRO, COTOVELO e PUNHO;
- Um bit informando se a garra está aberta ou fechada;
- Quatro bits referente as quatro chaves que estão instaladas no painel de comando;
- Um bit informando a condição do sensor capacitivo e um bit informando a condição atual do sensor indutivo.
- Informa o tipo de metal que está sendo recolhido pelo braço.

Ao todo o programa envia 22 bits de informações em forma de vetor para compor e transmitir o protocolo. Esse sinal é acionado por interrupção a uma frequência de 1KHz, porém dentro da interrupção existe um contador que permite a liberação dos dados para o supervisor a uma frequência de 10HZ. Sinais acima dessa velocidade geram acúmulo de informações no *buffer* da porta serial fazendo com que exista um alto *delay* nas transições de imagem. A figura 3.36 ilustra o sinal de comando referente a interrupção:

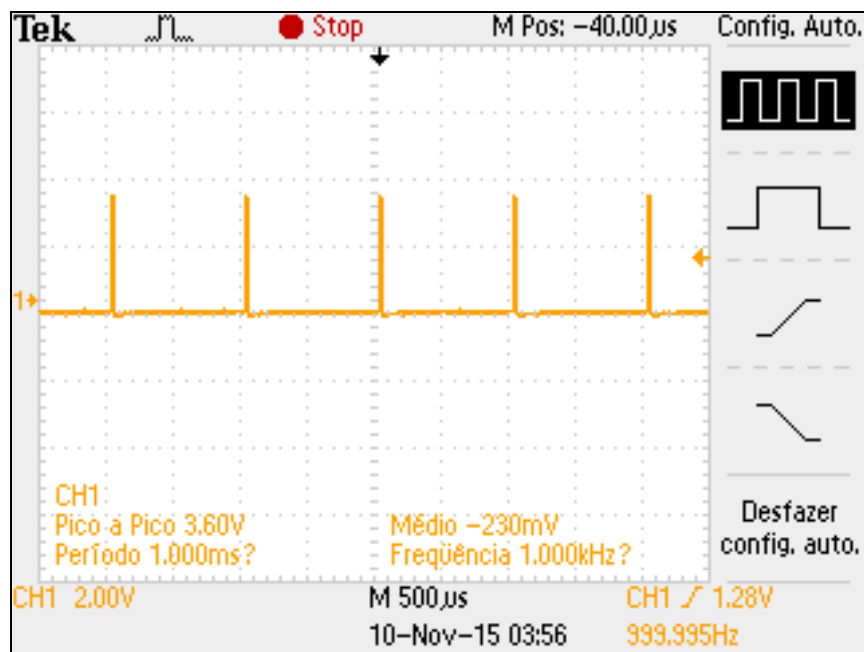


Figura 3.36 - Sinal PWM a 1KHz.

3.3 PLATAFORMA DE SUPERVISÃO

Foi desenvolvido um sistema supervisor para controle e monitoramento do braço mecânico, em linguagem LabVIEW®, juntamente com a *Toolbox Robotics*, ambos disponibilizados e distribuídos pela empresa *National Instruments*™.

3.3.1 Protocolo de Comunicação

A comunicação entre o microcontrolador e o LabVIEW se deu através de interface serial RS232, através de um pequeno protocolo de comunicação desenvolvido especificamente para este projeto. Convencionou-se uma tabela de equivalência dos números em letras, sendo toda a comunicação realizada na forma de algarismos semelhante ao código ASCII. A figura 3.37 demonstra o formato em que os dados são encapsulados para a transmissão.

Representação	Início	Base		Ombro			Cotovelo			Punho			Garra	A / M	I / P	H / D	Ref	CAP	IND	MOT	Tipo	Fim
Intervalo Válido	A	E-G	E-F	E-N	E-N	E-N	E-N	E-N	E-N	E-N	E-N	E-N	E-F	E-F	E-F	E-F	E-G	E-F	E-F	E-F	E-F	D
Posição	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21

Figura 3.37 - Encapsulamento de dados para comunicação serial.

Os dados encapsulados formam uma cadeia de 22 caracteres, respectivamente. Essa cadeia é armazenada dentro do microcontrolador em forma de vetor e enviada para a porta serial através dos sinais internos de interrupção do Arduino. Para as posições do OMBRO, COTOVELO e PUNHO são transmitidos três valores, representando as casas das centenas, dezenas e unidades do valor analógico enviado do potenciômetro, tornando possível recompor o número original. Para os demais dados transmitidos, trata-se de valores discretos, representados por um ou mais bits para informar ao supervisor o status de cada dispositivo monitorado. Neste caso, o motor da esteira, os sensores de presença, a posição da garra, os estados lógicos das chaves de comando e o tipo de peça que está sendo transportada.

Sendo assim, sempre que houver a necessidade de comunicação entre as plataformas física e de supervisão, os valores numéricos são convertidos em uma sequência de caracteres,

encapsulados e transmitidos, cabendo ao receptor interpretar as informações recebidas e recuperar os valores numéricos transmitidos.

Para evitar erros no processo de comunicação, foram utilizados dois marcadores, um para o início da transmissão e outro para fim da transmissão, para isto foram escolhidas as letras A e D respectivamente. O receptor permanece observando a porta serial, sempre que for recebido o identificador de início de transmissão, o processo de captação dos dados é iniciado, ao fim da coleta dos dados da porta serial, é feita a verificação do último dado transmitido, devendo este ser o identificador de fim de transmissão, caso contrário o LabVIEW interpretará que houve um erro na transmissão, descartando os dados coletados e aguardando nova transmissão.

A tabela 3.1 mostra a equivalência entre os algarismos ASCII e seus correspondentes numéricos no protocolo implementado.

Tabela 3.1 - Tabela e seus correspondentes numéricos.

ASCII	VALOR	ASCII	VALOR
=	0	N	9
A	99	O	10
B	-1	P	11
C	-1	Q	12
D	98	R	13
E	0	S	14
F	1	T	15
G	2	U	16
H	3	V	17
I	4	W	18
J	5	X	19
K	6	Y	20
L	7	Z	21
M	8		

Como mencionado anteriormente, o Arduino está programado para entrar na rotina de programação e uma frequência de 1KHz, dentro desta rotina é empregado um contador para o controle de envio de dados na porta serial. Em uma frequência de 10Hz, o vetor protocolo é enviado para a porta serial para atualização do supervisor. A figura 3.38 mostra o envio de dados na porta serial do microcontrolador alimentando o supervisor com os dados para atualização.

A comunicação entre as interfaces Processador-Supervisor é feita de forma simplex, onde uma vez recebido e interpretado os dados, o supervisor atualiza a interface gráfica, exibindo a posição atual do braço robótico, atualiza a contagem de peças manipuladas, indica se a esteira está em movimento e o estado atual dos sensores. A figura 3.39 apresenta um esquema do processo de comunicação entre as interfaces.

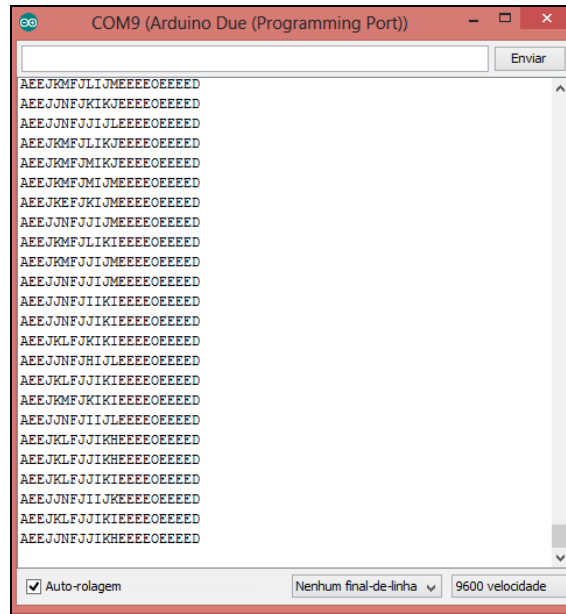


Figura 3.38 - Protocolo de comunicação.

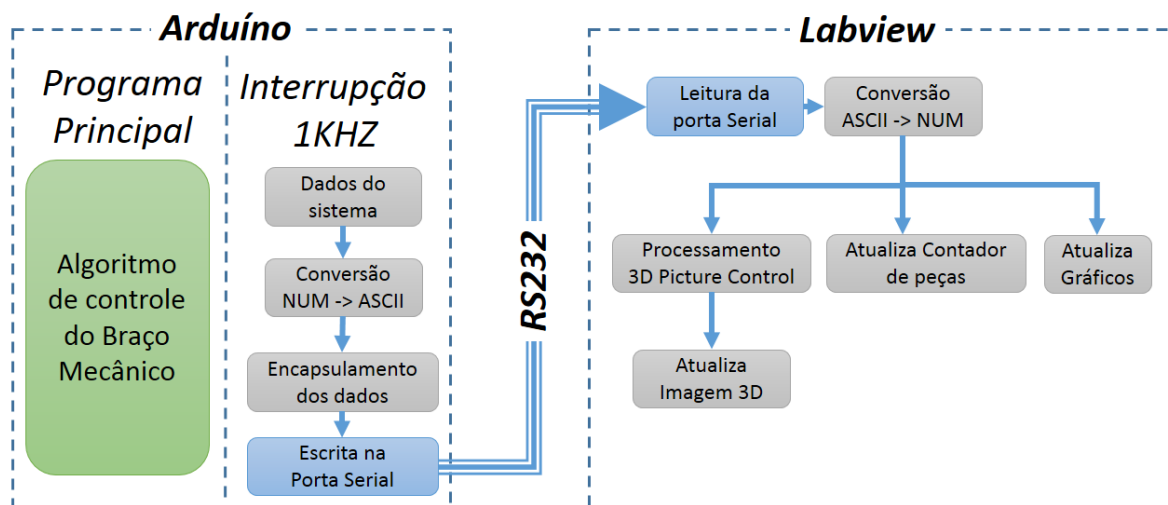


Figura 3.39 - Esquema de comunicação entre Arduino e LabVIEW.

3.3.2 Tela supervisão

A função de processamento da imagem tridimensional foi baseado no modelo desenvolvido pela *National Instruments* e consiste em uma série de manipulações da geometria, devendo a cada ciclo de leitura da porta serial, atualizar a posição e rotação de todos os sólidos geométricos que formam o braço virtual. Primeiramente os dados recebidos do Arduino são

reunidos em um vetor contendo a posição e estado de cada um dos quatro graus de liberdade do braço, a figura 3.40 apresenta o trecho do código fonte responsável pela criação do vetor de posição.

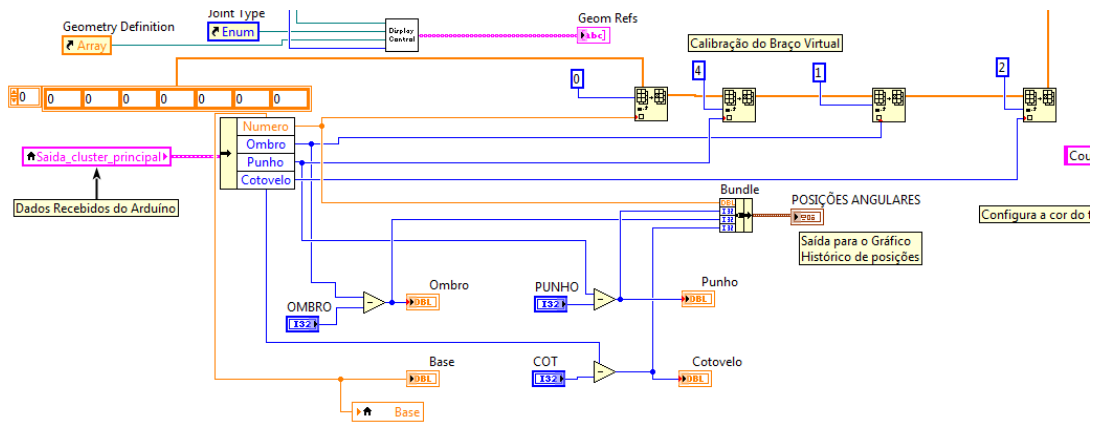


Figura 3.40 - Trecho responsável por informar as posições das juntas.

Ao final da criação do vetor, os dados são enviados a uma função que atualiza na tela a posição, rotação e escala dos blocos elementares que formam o braço virtual reproduzindo na tela uma animação que corresponde à posição real do braço físico. A representação do braço está ilustrado na figura 3.41.

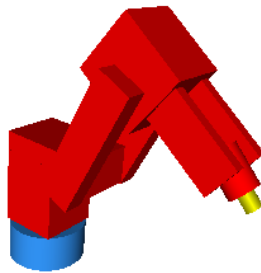


Figura 3.41 - Desenho 3D que representa o braço físico [20].

Uma das vantagens do LabVIEW é a capacidade de manipular objetos e links para criar LED's, chaves seccionadoras, barras de carregamento, ponteiros para supervisão analógica e uma infinidade de itens necessários para compor um sistema de supervisão completo a nível comercial e industrial.

A figura 3.42 representa a tela do supervisor desenvolvido, sendo este dividido em seis partes distintas, sendo um gráfico com o histórico da posição angular de cada um dos motores, uma figura tridimensional animada com a posição real e atual do braço físico, estado de alguns parâmetros operacionais, informações do referenciamento e calibração do braço, animação indicando o estado da esteira e um gráfico de bancada com o histórico de peças transmitidas e indicadores identificando o tipo de peça que está sendo transmitida.

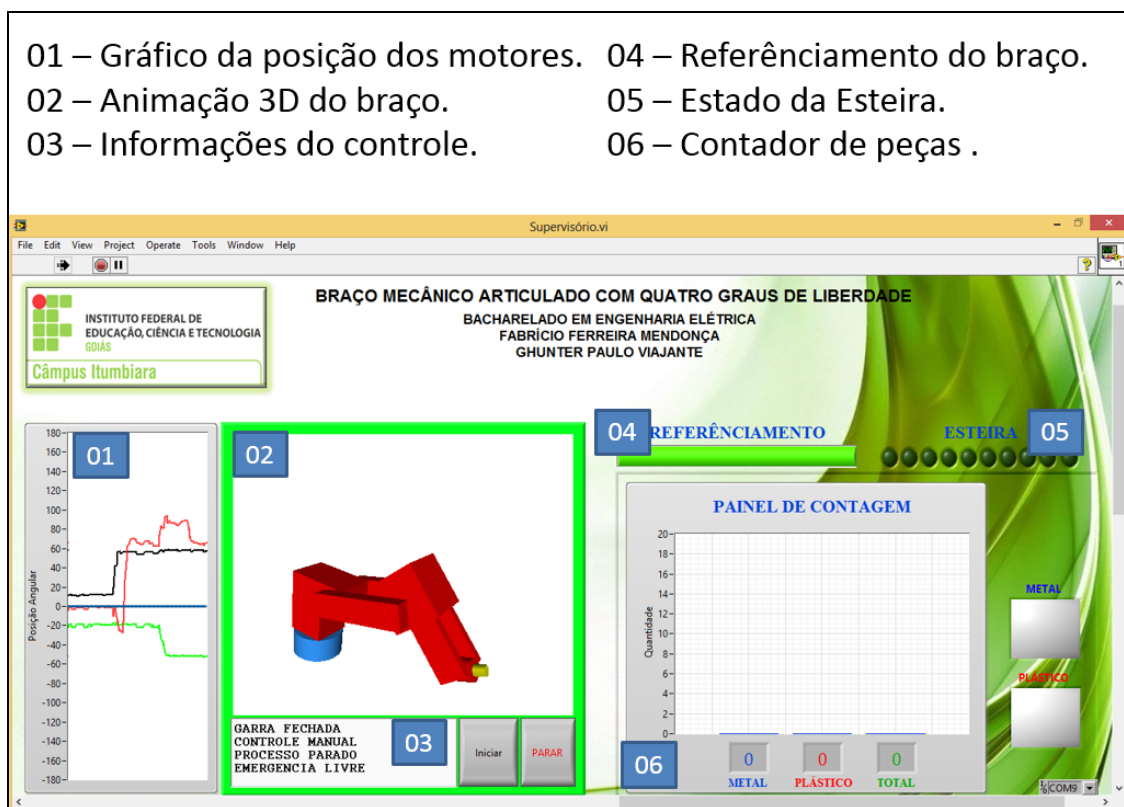


Figura 3.42 - Descrição dos itens da tela de supervisionamento.

CAPÍTULO 4

RESULTADOS EXPERIMENTAIS

A principal função do braço robótico deste projeto é separar peças metálicas de peças plásticas que são conduzidas através de uma esteira transportadora simulando muito bem as aplicações industriais que exigem manipulação de objetos com pouca mão de obra envolvida. Os robôs em processos industriais tem o foco em velocidade e continuidade de processos, principalmente quando eles são repetitivos.

Para esta projeto foi necessário utilizar, além do manipulador, alguns componente e equipamentos auxiliares para emular uma planta industrial. As esteiras utilizadas foram fornecidas pela instituição de ensino (IFG – Câmpus Itumbiara) juntamente com as peças e os sensores. As peças devem ser inseridas na esteira de forma manual e de maneira aleatória, pois o manipulador é quem vai separá-las. A peça possui uma massa de aproximadamente 200g percorre a esteira em uma velocidade aproximadamente de 4cm/s (0,04m/s no SI) e é paralisada quando a peça é identifica pelos sensores ao final do percurso. Esse conjunto de alimentação é ilustrado na figura a 4.1.



Figura 4.1 - Esteira de alimentação de peças.

4.1 REFERÊNCIAMENTO ESPACIAL DO MANIPULADOR

O primeiro passo ao energizar a fonte o microcontrolador é realizar a função de referenciamento para que o braço consiga se localizar e tomar como referencia suas posições espaciais. Essa medida foi adotada devido ao erro imposto pelo potenciômetro (utilizado como sensor de posição), que pode mudar o valor devido ao desgaste físico interno do componente.

Dessa forma é possível controlar manualmente o protótipo através das chaves seletoras no painel para colocar o braço na posição de descanso, de início ou qualquer outra atividade. Mas o objetivo deste passo é colocá-lo na posição em que o braço vai pegar a peça, ou seja, o braço deve ser manipulado manualmente até a posição em que a peça será retirada. A partir deste ponto é possível dar o comando para gravar a posição e estas coordenadas serão utilizadas na rotina do microcontrolador para calcular todas as outras posições como a coordenada da caixa onde será depositada a peça de metal, a caixa onde será depositada a peça de plástico e a posição de descanso quando aguardando alimentação de peça. Todas as manobras são visualizadas em um display de LCD.

Neste instante o supervisório indica que o sistema está parado para referenciamento, a tela fica vermelha e uma barra de *load* é visualizada para informar o status de referenciamento. Este funcionamento é ilustrado na figura 4.2.

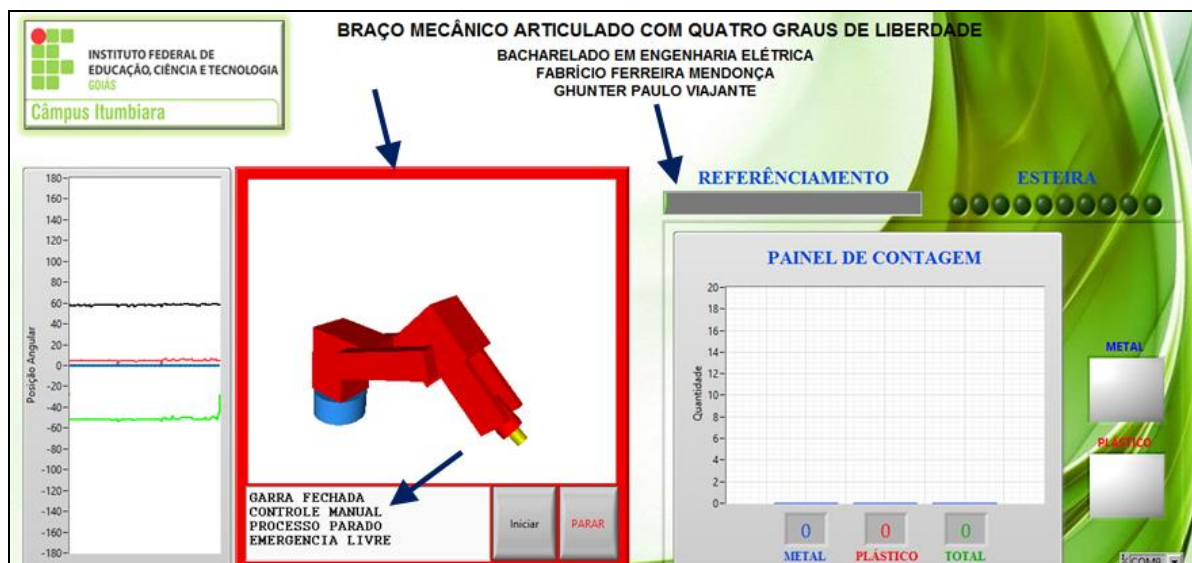


Figura 4.2 - Supervisório em estágio de referenciamento.

Sempre que o conjunto é desligado, reiniciado ou estabelecido uma nova conexão via porta serial, as informações armazenadas no passo de referenciamento são apagadas, sendo necessário calibrá-lo novamente em caso de funcionamento novamente. Essa medida trás como vantagem para o projeto em casos de situações de transporte ou locomoção do braço robótico, pois a posição de pegada do objeto pode sempre ser configurada de acordo com o lugar em que a esteira de alimentação será instalada. Caso o espaço seja muito pequeno é possível estabelecer como posição inicial uma nova coordenada, fazendo necessário neste caso, realizar uma nova rotina de referenciamento e então o manipulador está apto para trabalhar com a nova posição da esteira.

4.2 OPERAÇÃO E SUPERVISÃO

Uma vez referenciado, o protótipo está pronto para entrar em funcionamento. Nesse caso o mesmo deve ser estar selecionado para a opção automático impedindo que a rotina permita o funcionamento manual das articulações porque os mesmos botões utilizados para operar o braço em manual para referenciamento são os mesmo para coloca-lo em automático e habilitar as esteiras.

Não basta deixar o sistema selecionado em automático, isso apenas inibe a possibilidade de intervenções manuais, é preciso acionar a chave Iniciar/Parar para que a esteira entre em funcionamento e deixe o manipulador em pleno funcionamento e totalmente automático. A partir desta etapa só é possível parar o sistema selecionando novamente a chave Iniciar/Parar para que o braço pare de funcionar. Caso contrário, o manipulador continuará separando as peças que os sensores forem identificando na esteira.

Como o sistema é alimentado constantemente, o supervisor também informa a real situação física do manipulador. A figura 4.3 mostra o sistema acoplado em automático com o processo iniciado, tendo como referência de movimento os LED's que representam a esteira que ficam se movimentando para indicar que o processo está habilitado.

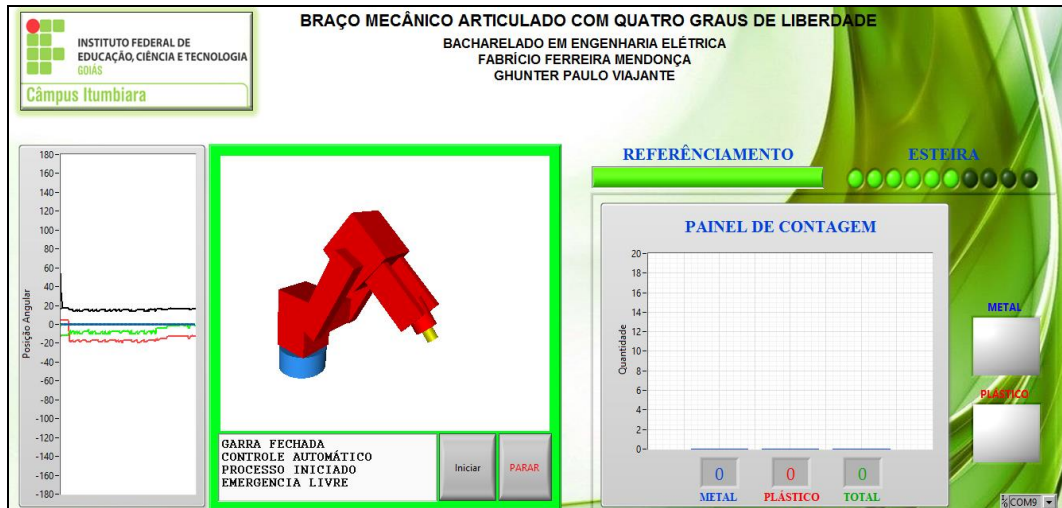


Figura 4.3 - Sistema habilitado aguardando peças.

Quando a peça é identificada, o supervisorio mostra qual tipo de peça o braço irá manipular e em seguida acrescenta no Pannel de Contagem a quantidade de peças que já foram selecionadas. Também, por motivo de segurança, é informado a situação da garra mostrando em forma de texto se a garra está aberta ou fechada. As informações são apresentadas na tela do supervisorio conforme ilustrado na figura 4.4.

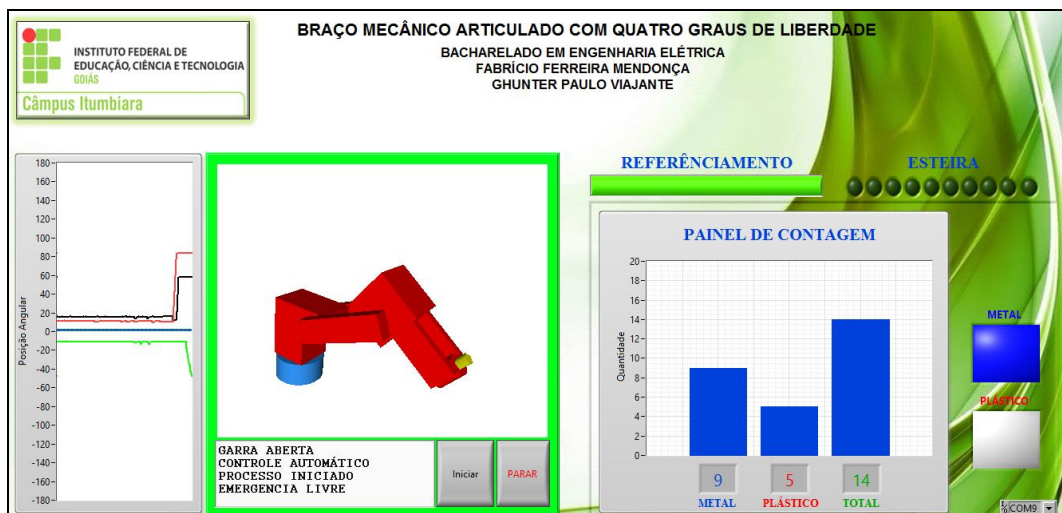


Figura 4.4 - Supervisorio mostrando histórico de peças.

A imagem 3D replica todos os movimentos realizados no espaço físico com o objetivo de informar o operador o que está acontecendo na planta, simulando o funcionamento industrial de

um braço robótico. Uma das funções que foram adotadas é o gráfico indicando os ângulos de cada articulação para efeito de visualização e histórico. A figura 4.5 ilustra o assunto tratado.

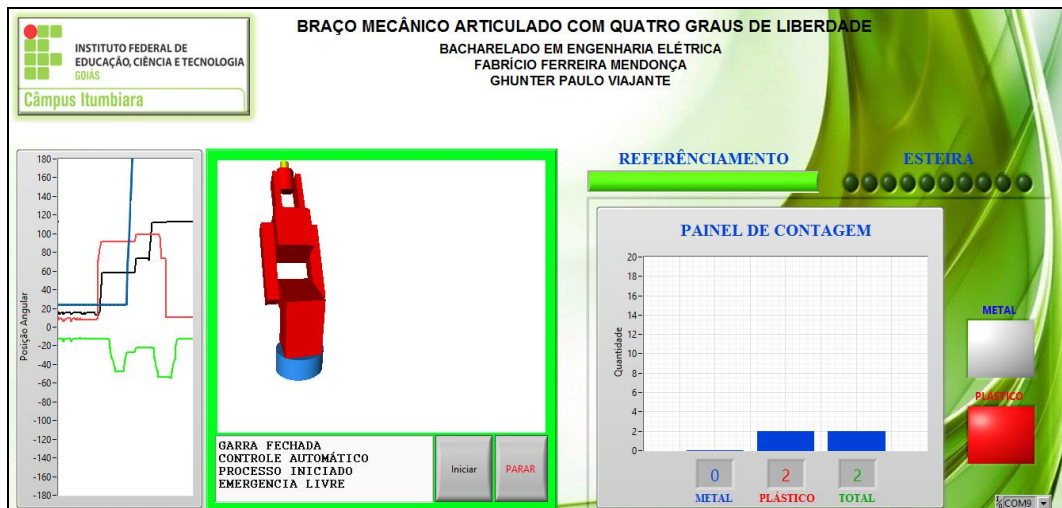


Figura 4.5 - Braço robótico em movimento.

4.3 CONTROLE DE POSIÇÃO DOS MOTORES

Os motores escolhidos para o projeto não tiveram as reduções de velocidade dimensionadas, pois conforme já relatado, os motores adquiridos foram motores de sucata a fim de reduzir ao máximo custo para elaboração do projeto. Para processos precisos quanto maior a relação de redução de velocidade das caixas de engrenagens melhor é o desempenho do sistema para controle de posição de eixo. Além de trazer ótimos benefícios em relação ao torque elevado que a caixa de engrenagem proporciona.

O controle posição adotado foi simples e barato baseado em uma malha fechada onde o programa identifica a posição do vínculo em relação ao setpoint e determina a velocidade que o motor irá funcionar de acordo com o ganho K_p estabelecido no código. Em uma situação onde a posição é igual ao setpoint a velocidade se torna zero realizado a parada do movimento.

Para cada junta, ou articulação, o motor se comporta de uma forma diferente devido a carga mecânica no qual ele está submetido e ao limite de velocidade imposto pelos próprios aspectos construtivos da máquina.

O sinal de leitura enviado através do potenciômetro é proporcional a posição assumida pelos motores na prática, e esse sinal se torna instável devido a interferência dos ruídos e de mal

contatos ocasionados pelo atrito entre o cursor e a trilha de carvão interna do potenciômetro. O aspecto construtivo demonstrado citado pode ser visualizado na figura 4.6.

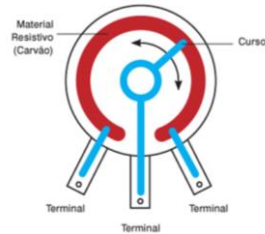


Figura 4.6 - Aspecto construtivo Potenciômetro.

A própria vibração do sistema afeta a capacidade de leitura, pois um simples movimento já altera a resistência do potenciômetro interferindo no sinal de tensão lido nos terminais analógicos do microcontrolador. Para diminuir a variação de leitura no Arduino, foi acrescentado no código um *Filtro Medias Móveis* que faz uma média das amostras dos sinais lidos durante um rápido intervalo de tempo. O filtro digital não é suficiente para garantir a precisão do dispositivo, mas é o bastante para uma boa referência de posição limitando o protótipo a movimentos mais abertos que não exigem muita precisão. A figura 4.7 ilustra o acompanhamento do controle de posição para as articulações do OMBRO, COTOVELO e PUNHO para mostrar o valor da saída em relação ao setpoint desejado.

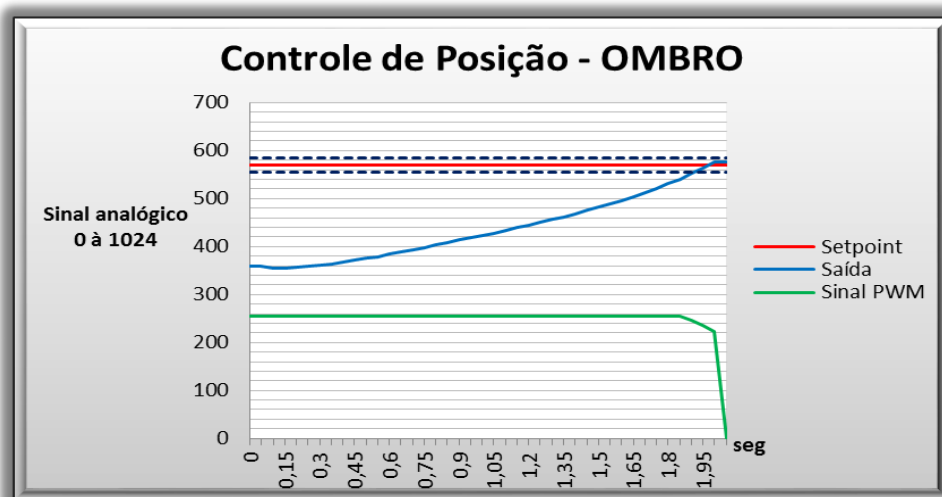


Figura 4.7 - Setpoint X Saída - OMBRO.

Um sinal analógico equivalente a 255 representa um sinal PWM de 100%. Nota-se que devido ao OMBRO ser a junta mais sobrecarregada, mecanicamente falando, ela exige um alto torque durante um longo tempo de execução e para propiciar esse torque desejado foi preciso injetar praticamente toda potência disponível pela fonte.

O controle de posição referente ao COTOVELO pode ser visualizado na figura 4.8.

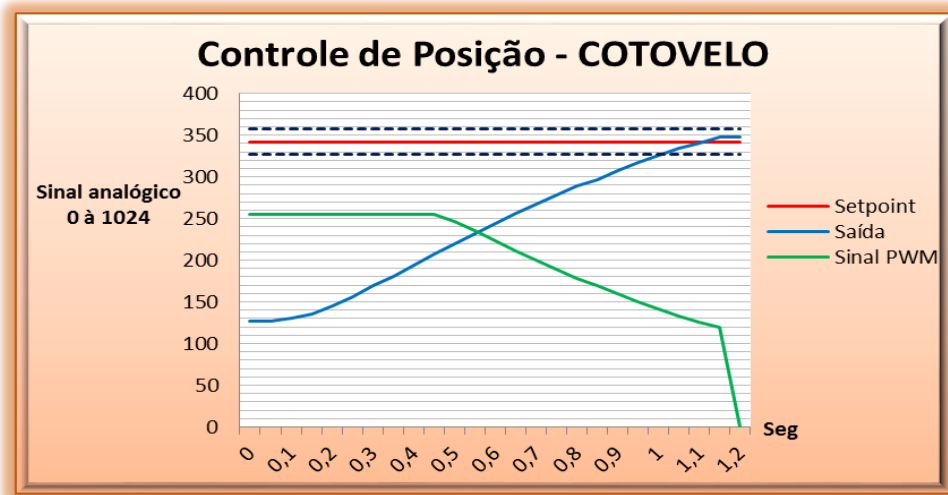


Figura 4.8 - Setpoint X Saída - COTOVELO.

A articulação do COTOVELO, por ser mais leve, consegue atingir o setpoint de forma suave junto com o decaimento da velocidade e do sinal de acionamento PWM. Essa transição ocorre de acordo com que o sinal de saída vai se aproximando do Setpoint. A figura 4.9 mostra uma variação na largura de pulsos do sinal PWM ilustrando o controle de velocidade para suavizar os movimentos.

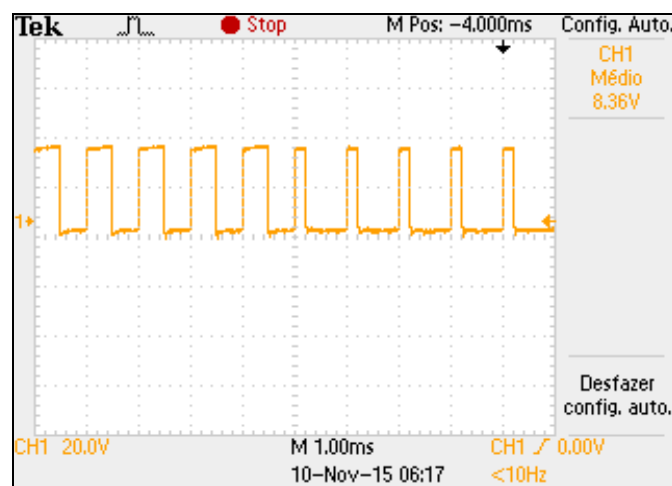


Figura 4.9 - Redução da largura de pulso.

Uma particularidade do motor responsável por movimentar o PUNHO é o peso da garra em relação ao redutor do motor, pois se instalado um motor muito grande ele se tornaria um peso para a estrutura que compõe o vínculo entre o COTOVELO e o PUNHO. Nisso, quando o motor é desenergizado, o peso da garra consegue vencer o momento de inercia do redutor resultando em movimento inesperado. Diante dessa situação, quando o setpoint é alcançado, ainda é necessário deixar uma tensão residual nos enrolamentos do motor para equilibrar as forças entre o torque de subida e o peso da garra que tende a movimentar a junta, devido a isso a largura de pulso PWM é mantida por um valor fixo no gráfico “Setpoint X Saída - PUNHO”. Procedimento ilustrado na figura 4.10.

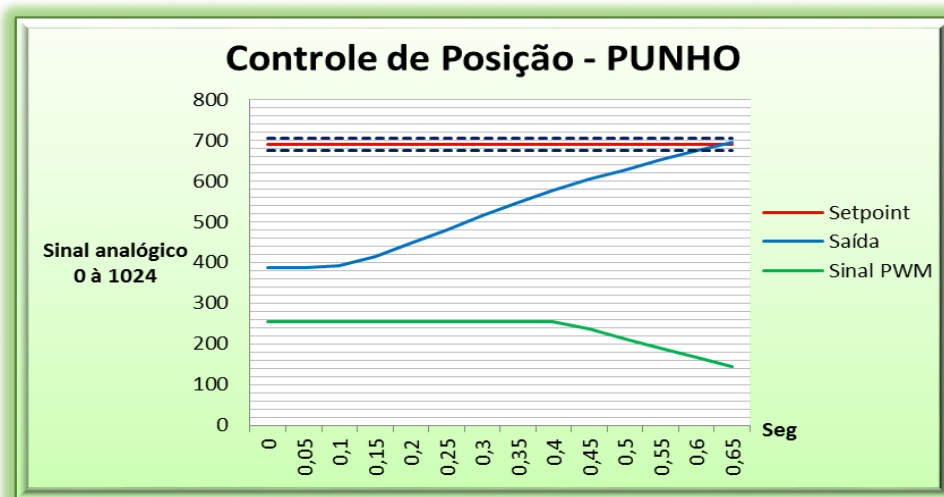


Figura 4.10 - Setpoint X Saída - PUNHO.

4.4 ALCANCE DE ATUAÇÃO DO BRAÇO

Após construído e realizado os testes necessários de funcionamento e operação, foi realizado alguns movimentos aleatórios para retirada de valores em relação ao alcance que o braço robótico possui. O manipulador completamente esticado consegue atingir uma distância de aproximadamente 700mm, equivalente a 0,7 metros no sistema SI. Os cabos de alimentação dos motores e dos sensores limitam o movimento rotacional da cintura em 270°, uma medida que pode atingir 360° caso os cabos sejam estendidos. Como é um movimento desnecessário, esta manobra não foi realizada. As medidas dos vínculos estão ilustrados na imagem 4.11.

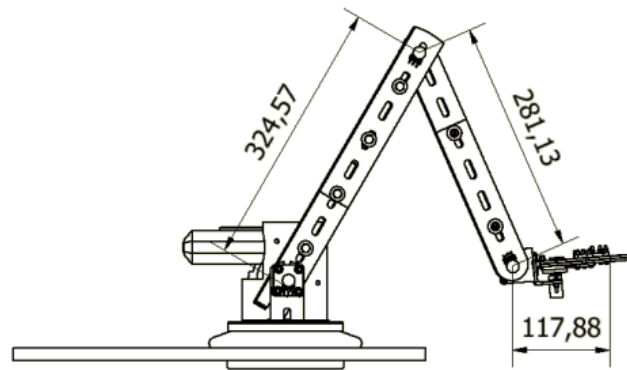


Figura 4.11 - Medidas Vínculos.

A cintura consegue atingir posições variadas em uma faixa de movimento de até 270° em um raio limitado de 710mm conforme visto na figura 4.12.

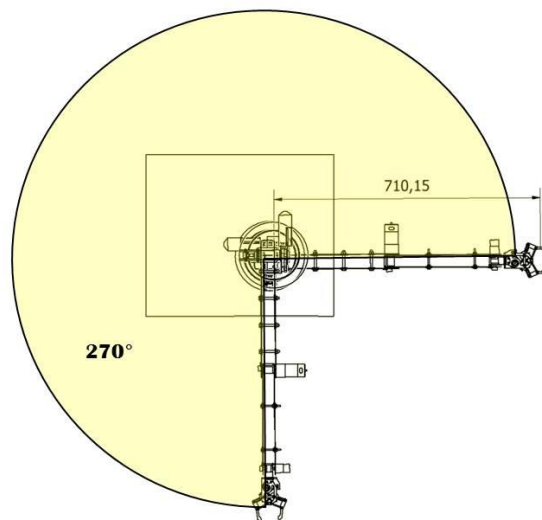


Figura 4.12 - Capacidade rotativa Cintura.

O OMBRO possui seu movimento limitado a 90° , conforme informado na figura 4.13, devido aos aspectos construtivos desta articulação. O movimento rotativo acima desta media não é permitida devido as limitações mecânicas existentes da carcaça do motor de acionamento da junta que não permite o vínculo ultrapassar esta escala.

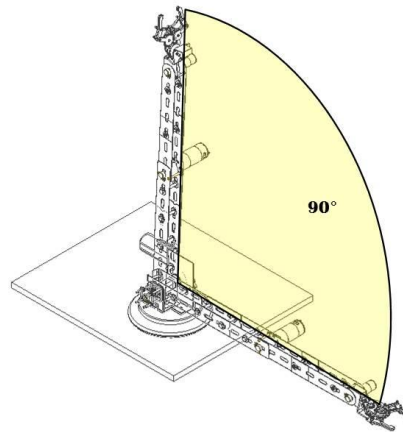


Figura 4.13 – Abertura OMBRO em 90°.

Para o COTOVELO foi estabelecido um ângulo máximo de 200° para não comprometer os limites do potenciômetro, sendo este também, uma medida angular satisfatória para o projeto ilustrada na figura 4.14.

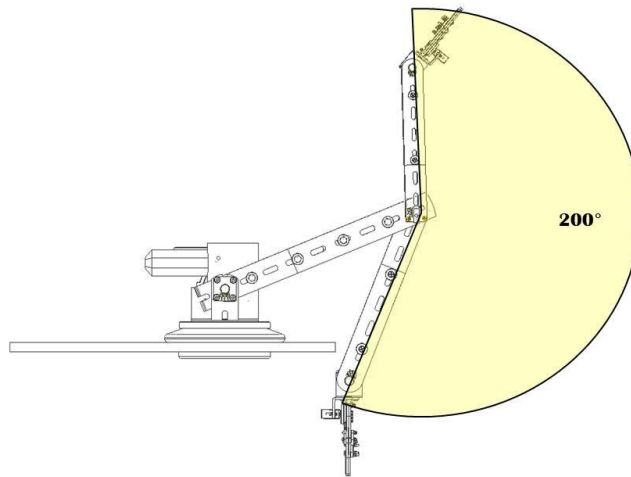


Figura 4.14 - Liberdade COTOVELO.

Pelas mesmas condições do motor e vínculo do COTOVELO, o motor de acionamento da junta do PUNHO também possui uma liberdade de 200° de movimento, informado da figura 4.15;

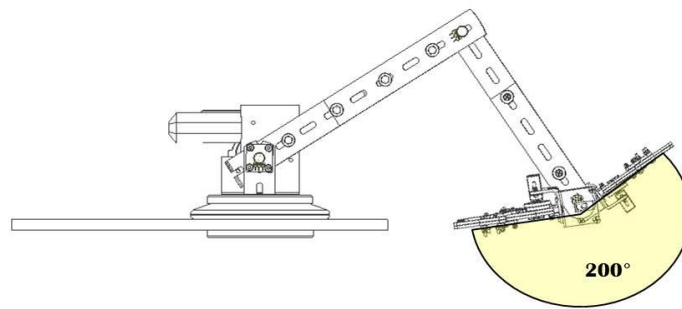


Figura 4.15 - Liberdade PUNHO.

Ao final do projeto, o manipulador foi submetido a diferentes testes para verificar a capacidade de manipulação de peças referente ao peso total que o braço robótico consegue transportar. Em circunstâncias normais de operação, o protótipo conseguiu erguer peças com uma massa de aproximadamente 500g.

Para corpos mais pesados e de maiores dimensões, é necessário atribuir mais dispositivos mecânicos para melhor desempenho como inserção de contra pesos ou de molas helicoidais para melhorar a tração de subida do braço robótico.

4.5 LEVANTAMENTO DE CUSTOS

Inicialmente foi realizado um pré-levantamento da relação dos materiais a serem utilizados no projeto, desde os componentes eletrônicos até aos detalhes e acabamentos da estrutura final. O foco foi projetar um equipamento que permitisse a construção de um manipulador mecânico fazendo uso do mínimo de recursos possíveis para demonstrar que o projeto pode ser concretizado de forma simples e barata. Sempre priorizando dispositivos de baixo custo ou de custo algum, como aconteceu com o desenvolvimento da estrutura (vínculos) utilizando sobras de eletrocalhas, encontradas na sucata industrial da cidade, que tem finalidade de acondicionamento de cabos elétricos em recintos industriais e comerciais.

Após finalização do experimento também foi levantado o custo que seria gasto caso o intuito fosse desenvolver um manipulador mecânico de excelente desempenho para atender os padrões industriais de precisão de movimento. Os componentes mais caros seriam os motores para acionamento das juntas e do dispositivo de controle que pode atingir valores acima de R\$200,00 dependendo da versão. O levantamento feito se baseia na utilização de motores de

passo para substituir os motores de corrente contínua improvisados. Para substituir os motores da Cintura (BASE) e do OMBRO seria necessário fazer uso de motores com o torque acima de 100kgf.cm, pois estes teriam a capacidade de suprir a exigência mecânica que os outros não conseguiram. De acordo com a MABUCHI, fabricante do motor de vidro elétrico de carro, o mesmo possui um torque nominal de 10N.m, equivalente a 100Kgf.cm.

A previsão de gastos utilizando motores de passo e acoplamentos específicos chega atingir valores acima de 3,5 vezes mais do que o valor gasto no protótipo desenvolvido. Pode ser escolhido também servomotores ao invés de motores de passo, porém o custo com os drives de acionamento elevam ainda mais o preço do conjunto. Servomotores são máquinas precisas, porém possuem um alto custo de aquisição e a tensão de alimentação possui valores específicos dependendo da aplicação. Isso dificulta a acessibilidade ao material.

A seguir serão apresentadas duas planilhas contemplando a descrição dos materiais elétricos e mecânicos utilizados no trabalho. Uma tabela mostra o valor gasto para desenvolver o manipulador e a outra marca os componentes que deveriam ser substituídos para conseguir o projeto de alta performance, com melhor precisão, velocidade e rendimento. A tabela 4.1 informa o valor gasto no projeto.

Os materiais que não apresentam custo foram fornecidos pela instituição de ensino, outros encontrados em sucatas e alguns doados por alunos e professores. O Arduino DUE, por exemplo, foi um dispositivo fornecido pelo orientador do projeto enquanto a esteira transportadora e os sensores de presença foram disponibilizados pelo IFG – Câmpus Itumbiara. A tabela 4.2 mostra o levantamento de custo em caso de utilização dos motores de passo.

Tabela 4.1 - Valor gasto no projeto.

LEVANTAMENTO DE CUSTO					
Item	Descrição	Qtd	Unit. [R\$]	Total [R\$]	
1	Resistor Carbono 5%	47	R\$ 0,10	R\$	4,70
2	Potenciômetro Linear 10K	3	R\$ 1,60	R\$	4,80
3	Chave Alavanca 12V	4	R\$ 2,50	R\$	10,00
4	Chave tipo Push Button	1	R\$ 1,00	R\$	1,00
5	Comutador Wireless 315MHz	1	R\$ 55,00	R\$	55,00
6	Diodo 5A 60V	8	R\$ 1,40	R\$	11,20
7	Diodo 1,5A 50V	11	R\$ 0,20	R\$	2,20
8	Diodo Zenner 1N4728	2	R\$ 0,25	R\$	0,50
9	Transistor BC337	10	R\$ 0,26	R\$	2,60
10	Transistor TIP122/127	8	R\$ 1,30	R\$	10,40
11	Transistor TIP142/147	8	R\$ 3,50	R\$	28,00
12	Fios Jumper	40	R\$ 0,30	R\$	12,00
13	Transistor 2SC2073 1,5A	9	R\$ 1,00	R\$	9,00
14	Borne CBM 02 Terminais	11	R\$ 0,88	R\$	9,68
15	Borne CBM 03 Terminais	23	R\$ 1,80	R\$	41,40
16	Alojamento 12 Terminais	4	R\$ 0,80	R\$	3,20
17	Borne de Conexão Externo	12	R\$ 2,00	R\$	24,00
18	Fusível Vidro 10A	5	R\$ 0,60	R\$	3,00
19	Tela LCD 16x4 5V	1	R\$ 60,00	R\$	60,00
20	LED Emissor Infravermelho TIL 32	2	R\$ 0,80	R\$	1,60
21	LED Receptor Infravermelho TIL 78	2	R\$ 0,80	R\$	1,60
22	Protoboard 550 pontos	1	R\$ 53,00	R\$	53,00
23	Dissipador de Calor 180847	8	R\$ 2,10	R\$	16,80
24	Dissipador de Calor 183013	8	R\$ 2,50	R\$	20,00
25	Garra Robótica Alum.MK-II	1	R\$ 50,00	R\$	50,00
26	Fita LED Branca 12V	1	R\$ 35,00	R\$	35,00
27	Motor CC 12V - Vidro El. (OMBRO)	1	R\$ 30,00	R\$	30,00
28	Motor CC 12V - Limpador Pára-brisa (BASE)	1	R\$ 50,00	R\$	50,00
29	Motor CC 12V (COTOVELO)	1	R\$ 53,00	R\$	53,00
30	Servomotor Garra	1	R\$ 54,00	R\$	54,00
31	Base madeira	1	R\$ 100,00	R\$	100,00
32	Tinta Spray	2	R\$ 15,00	R\$	30,00
33	Organizador de cabos	10	R\$ 1,00	R\$	10,00
34	Acrílico de Proteção	2	R\$ 35,00	R\$	70,00
35	Confecção da Placa Eletrônica	1	R\$ 90,00	R\$	90,00
36	Cabo Comando 1mm ²	1	R\$ 20,00	R\$	20,00
37	Caixa papelão para armazenamento	1	R\$ 14,00	R\$	14,00
38	Parafusos e brocas	1	R\$ 45,00	R\$	45,00
39	Mola tração	1	R\$ 3,00	R\$	3,00
SEM CUSTO					
40	Arduino DUE	1	R\$ -	R\$	-
41	Sensor Capacitivo 22mm 12V	1	R\$ -	R\$	-
42	Sensor Indutivo 22mm 12V	1	R\$ -	R\$	-
43	Esteira de Alimentação de peças	1	R\$ -	R\$	-
44	Corpos de Prova Metal e Não Metal	10	R\$ -	R\$	-
45	Fonte 12/5/3,3V	2	R\$ -	R\$	-
46	Base Rotativa Plástico (CINTURA)	2	R\$ -	R\$	-
47	Estrutura Mecânica (VÍNCULOS)	5	R\$ -	R\$	-
48	Acoplamentos Tecnil	3	R\$ -	R\$	-
49	Chapa Aço Carbono 200x200x3mm	1	R\$ -	R\$	-
50	Motor CC 12V (PUNHO)	1	R\$ -	R\$	-
			Componentes	Total	
			283	R\$ 1.039,68	

*Valores baseados na compra no ano de 2015 com a cotação do dólar à R\$3,87.

Tabela 4.2 - Previsão de gasto com motor de passo.

LEVANTAMENTO DE CUSTO					
Item	Descrição	Qtd	Unit. [R\$]	Total [R\$]	
1	Resistor Carbono 5%	47	R\$ 0,10	R\$	4,70
2	Potenciômetro Linear 10K	3	R\$ -	R\$	-
3	Chave Alavanca 12V	4	R\$ 2,50	R\$	10,00
4	Chave tipo Push Button	1	R\$ 1,00	R\$	1,00
5	Comutador Wireless 315MHz	1	R\$ 55,00	R\$	55,00
6	Diodo 5A 60V	8	R\$ 1,40	R\$	11,20
7	Diodo 1,5A 50V	11	R\$ 0,20	R\$	2,20
8	Diodo Zenner 1N4728	2	R\$ 0,25	R\$	0,50
9	Transistor BC337	10	R\$ 0,26	R\$	2,60
10	Transistor TIP122/127	8	R\$ 1,30	R\$	10,40
11	Transistor TIP142/147	8	R\$ 3,50	R\$	28,00
12	Fios Jumper	40	R\$ 0,30	R\$	12,00
13	Transistor 2SC2073 1,5A	9	R\$ 1,00	R\$	9,00
14	Borne CBM 02 Terminais	11	R\$ 0,88	R\$	9,68
15	Borne CBM 03 Terminais	23	R\$ 1,80	R\$	41,40
16	Alojamento para conector 12 Terminais	4	R\$ 0,80	R\$	3,20
17	Borne de Conexão Externo	12	R\$ 2,00	R\$	24,00
18	Fusível Vidro 10A	5	R\$ 0,60	R\$	3,00
19	Tela LCD 16x4 5V	1	R\$ 60,00	R\$	60,00
20	LED Emissor Infravermelho TIL 32	2	R\$ -	R\$	-
21	LED Receptor Infravermelho TIL 78	2	R\$ -	R\$	-
22	Protoboard 550 pontos	1	R\$ -	R\$	-
23	Dissipador de Calor 180847	8	R\$ 2,10	R\$	16,80
24	Dissipador de Calor 183013	8	R\$ 2,50	R\$	20,00
25	Garra Robótica Alum.MK-II	1	R\$ 50,00	R\$	50,00
26	Fita LED Branca 12V	1	R\$ 35,00	R\$	35,00
27	Motor CC 12V - Vidro Elétrico 100Kgf.c	1	R\$ 570,00	R\$	570,00
28	Motor CC 12V - Limpador Para-brisa (BA	1	R\$ 570,00	R\$	570,00
29	Motor CC 12V (COTOVELO)	1	R\$ 190,00	R\$	190,00
30	Servomotor Garra	1	R\$ 54,00	R\$	54,00
31	Base madeira	1	R\$ 100,00	R\$	100,00
32	Tinta Spray	2	R\$ 15,00	R\$	30,00
33	Organizador de cabos	10	R\$ 1,00	R\$	10,00
34	Acrílico de Proteção	2	R\$ 35,00	R\$	70,00
35	Confeção da Placa Eletrônica	1	R\$ 90,00	R\$	90,00
36	Cabo Comando 1mm²	1	R\$ 20,00	R\$	20,00
37	Caixa papelão para armazenamento	1	R\$ 14,00	R\$	14,00
38	Parafusos e brocas	1	R\$ 45,00	R\$	45,00
39	Mola tração	1	R\$ -	R\$	-
SEM CUSTO					
40	Arduino DUE	1	R\$ 275,00	R\$	275,00
41	Sensor Capacitivo 22mm 12V	1	R\$ -	R\$	-
42	Sensor Indutivo 22mm 12V	1	R\$ -	R\$	-
43	Esteira de Alimentação de peças	1	R\$ -	R\$	-
44	Corpos de Prova Metal e Não Metal	10	R\$ -	R\$	-
45	Fonte 500W 12/5/3,3V	1	R\$ 150,00	R\$	150,00
46	Base Rotativa Plástico (CINTURA)	2	R\$ 150,00	R\$	300,00
47	Estrutura Mecânica (VÍNCULOS)	5	R\$ 60,00	R\$	300,00
48	Acoplamentos Tecnil	3	R\$ 95,00	R\$	285,00
49	Chapa Aço Carbono 200x200x3mm	1	R\$ 40,00	R\$	40,00
50	Motor CC 12V (PUNHO)	1	R\$ 137,00	R\$	137,00
			Componentes		Total
			278		R\$ 3.659,68

CAPÍTULO 5

CONSIDERAÇÕES FINAIS

O escopo proposto de construção e funcionamento do braço robótico foi concluído com sucesso. A estrutura mecânica, apesar de improvisada, demonstrou possuir uma boa resistência física em relação aos materiais que são disponibilizados prontos no mercado da robótica educacional.

O circuito elétrico é estável e suporta altos níveis de corrente e um bom desempenho na proteção em caso de travamentos ou curto-circuito. Durante a fase experimental dois fusíveis de 10A foram substituídos devido a imprevistos ocorridos no sistema eletrônico proporcionados na fase operacional. Mesmo em caso de falha dos fusíveis de proteção, a fonte utilizada possui dispositivos contra corrente de surto limitando a potência disponível da saída do alimentador.

A precisão dos movimentos não ficaram compatíveis a um manipulador industrial, assim como já era esperado devido as incertezas geradas pelo sinal do potenciômetro, um sensor improvisado que não possui características construtivas suficientes para esta finalidade.

Mesmo com toda a dificuldade enfrentada, decorrente das falhas ocorridas nos sinais de referência de posição, o experimento funcionou de maneira gratificante oferecendo excelentes resultados e sendo muito bem recebido pela comunidade docente podendo ser utilizado em folders, Banners e objeto de divulgação do curso em seminários e palestras que envolvem inovação e tecnologia. Para fins acadêmicos ele possui uma excelente finalidade, pois é possível estudar e aplicar detalhadamente muitas disciplinas que foram ofertadas ao longo de todo o curso.

Ademais, destaca-se o desenvolvimento de um aplicativo supervisório que foi desenvolvido especialmente para atender as necessidades do projeto, sendo possível controlar e supervisionar o braço robótico de forma remota e visualizar os movimentos realizados pelo manipulador de forma virtual, através da imagem 3D disponível supervisório.

O projeto foi apresentado III Fórum Mundial de Educação Profissional e Tecnológica que ocorreu na cidade de Recife (PE), no período de 26 a 29 de Maio de 2015, no stand do IFG -

Goiás tendo oportunidade de demonstrar o seu funcionamento no espaço reservado para o MEC. Convite realizado pelos organizadores do evento ao se interessarem pelo projeto.

5.1 SUGESTÃO DE TRABALHOS FUTUROS

Deixa-se como recomendação para projetos futuros a substituição do elemento sensorial de posição por encoders incrementais ou absolutos que são vendidos em várias faixas de preços e fornecem diversos modelos para diferentes aplicações.

É possível aperfeiçoar os movimentos executados pelo manipulador implementando outras técnicas de controle para melhorar a precisão e a repetibilidade através do controle de posição dos motores.

Estabelecer outra forma de indicação e apuração de coordenadas fazendo uso das cinemáticas direta e inversa. Além de que é possível modificar a estrutura mecânica do manipulador para obter classificações geométricas diferentes, podendo trabalhar com outros tipos de coordenadas, como as coordenadas polares ou cilíndricas.

Uma sugestão também interessante é fazer a substituição dos motores de corrente contínua por motores de passo, pois são motores relativamente leves, com um alto nível de desempenho mecânico e o controle de posição pode ser feito através da contagem dos pulsos emitidos pelo dispositivo controlador, sem existir a necessidade de utilizar sensores acoplados no eixo.

APENDICE A – CÓDIGO FONTE

Abaixo será possível visualizar o código fonte utilizada no microcontrolador.

```
#include <LiquidCrystal.h>
//#include <Servo.h>
//Servo GARRA;

int leituraOMBRO, leituraCOTOVELO, leituraPUNHO, INICIA_PARA, SENSOR_CAP,
SENSOR_IND, AUTO_MAN, REFERENCIAMENTO, CALIBRACAO, HAB_DES, pulsos,
ref, comando;

int OMBRO_P1, COTOVELO_P1, PUNHO_P1, OMBRO_P2, COTOVELO_P2, PUNHO_P2,
OMBRO_P3, COTOVELO_P3, PUNHO_P3, ganho;

int tempo, porta_serial, conta_bit, indica, contagem, tipo, cont_interrupt, i, lcd_cont,
conta_interrupcao, METAIS, PLASTICOS;

char protocolo[21]; // relacionado com a funcao "dados"

int REF_CONT = 0;

LiquidCrystal lcd(46, 44, 42, 40, 38, 36);

void posicaoP1();
void posicaoP3(int pos);
void fecha_garra();
void sobe_cotovelo(int pos, int pwm);
void sobe_punho(int pos, int pwm);
void sobe_ombro(int pos, int pwm);
void base_direita(int pwm, int pos);
void desabilita_portas();
void posicaoINI(int pos);
void converte_pot(int cod, int casa);
void converte_base(int cod, int casa);
void converte_dig(int sinal, int casa);

void posicaoP2(int pos);
void abre_garra();
void gravar();
void desce_cotovelo(int pos, int pwm);
void desce_punho(int pos, int pwm);
void desce_ombro(int pos, int pwm);
void base_esquerda(int pwm, int pos);
void leituras();
void teste();
void posicaoP3();
void referencia();
void dados();
```

```
void atualiza_lcd();                void rotina_maluca();
void calibracao();
```

```
char troca(int aux);
```

```
//=====
```

```
void setup()
{
```

```
pinMode(2, OUTPUT);    //SOBE PUNHO
pinMode(3, OUTPUT);    //DESCE PUNHO
```

```
pinMode(4, OUTPUT);    //SOBE COTOVELO
pinMode(5, OUTPUT);    //DESCE COTOVELO
```

```
pinMode(6, OUTPUT);    //SOBE OMBRO
pinMode(7, OUTPUT);    //DESCE OMBRO
```

```
pinMode(8, OUTPUT);    //BASE DIREITA
pinMode(9, OUTPUT);    //BASE ESQUERDA
```

```
pinMode(10, OUTPUT);    //GARRA
//GARRA.attach(10,600,2200); //GARRA
```

```
pinMode(13, OUTPUT);    //Pino especifico para interrupção
```

```
pinMode(47, OUTPUT);    //LED
pinMode(49, OUTPUT);    //ESTEIRA ALIMENTAÇÃO
pinMode(51, OUTPUT);    //ESTEIRA SAIDA 1
pinMode(53, OUTPUT);    //ESTEIRA SAIDA 2
```

```
pinMode(52, INPUT);    //ROTINA DE REFERENCIAMENTO
pinMode(23, INPUT);    //INICIA/PARA
pinMode(25, INPUT);    //AUTO/MAN
pinMode(27, INPUT);    //SENSOR INDUTIVO
pinMode(29, INPUT);    //SENSOR CAPACITIVO
pinMode(31, INPUT);    //REFERENCIAMENTO
pinMode(33, INPUT);    //HAB/DES
```

```
lcd.begin(16, 4); //Indica o tipo de display ( pin 46 --- RS  pin 44 --- E  42 --- D4  40 --- D5
38 --- D6  36 --- D7 )
```

```
attachInterrupt(24, enviar_dados, RISING); //Configura interrupção ( nº da porta que recebe o
sinal , chama função , RISING pega borda de subida )
```

```
Serial.begin(9600);
```

```
tipo = 0; comando = 0; cont_interrupt = 0; lcd_cont = 0; conta_interrupcao = 0; METAIS = 0; P
LASTICOS = 0;
```

```
ganho = 20; tempo = 100;
```

```
}
```

```
//-----LOOP-----LOOP-----LOOP-----LOOP-----LOOP-----
```

```
void loop()
```

```
{
```

```
digitalWrite(47, HIGH); //Liga LED's
```

```
analogWrite(13 , 10); //Dispara sinal pra interrupção
```

```
leituras();
```

```
desabilita_portas();
```

```
if(AUTO_MAN == HIGH)//Seleciona Automático
```

```
{
```

```
Serial.print(" "); //Mantém a porta serial ocupada para não alterar bits de forma errada
```

```
// if (INICIA_PARA == HIGH){ comando = 1; delay(1100); } //Recebe sinal do controle/chave  
e faz o selo para partida da rotina
```

```
while(INICIA_PARA == HIGH)//Autoriza Inicio
```

```
{
```

```
digitalWrite(49, HIGH); //Liga Esteira Alimentação
```

```
Serial.print(" "); //Mantém a porta serial ocupada para não alterar bits de forma errada
```

```
converte_dig(HIGH , 19); // Converte MOTOR para colocar no vetor de protocolo
```

```
leituras();
```

```
if(SENSOR_IND == HIGH) //Metal
```

```
{
```

```
delay(1485); // Tempo para parar esteira
```

```
digitalWrite(49, LOW); //Desliga Esteira Alimentação
```

```
converte_dig(LOW , 19); // Converte MOTOR para colocar no vetor de protocolo
```

```
METAIS = METAIS + 1; // Incrementa na quantidade
```

```
if(METAIS > 10) { METAIS = 0; PLASTICOS = 0;}// Zera contagem quando excedo o  
alfabeto
```

```
protocolo[18] = troca(METAIS); // Converte para inserir no protocolo
```

```
tipo = 1; // Metal
```

```
protocolo[20] = troca(tipo); // Insere no protocolo
```

```
posicaoP1();
```

```
    fecha_garra();
    delay(tempo);
    posicaoP3( 3 ); // CAIXA
    posicaoINI( 3 );

    }//Metal
//
if(SENSOR_CAP == HIGH)//Plástico
{
    delay(1100); //Tempo para parar esteira
    digitalWrite(49, LOW); //Desliga Esteira Alimentação
    converte_dig(LOW , 19); // Converte MOTOR para colocar no vetor de protocolo

    PLASTICOS = PLASTICOS + 1; // Incrementa na quantidade
    if(PLASTICOS > 10) { METAIS = 0; PLASTICOS = 0;}// Zera contagem quando excedo
o alfabeto
    protocolo[17] = troca(PLASTICOS); // Converte para inserir no protocolo
    tipo = 2; // Plastico
    protocolo[20] = troca(tipo); // Insere no protocolo

    posicaoP1();
    digitalWrite(51, HIGH); // LIGA ESTEIRA DE SAIDA
    fecha_garra();
    delay(tempo);
    posicaoP2( 5 ); // ESTEIRA
    posicaoINI( 5 );

    }//Plástico

    tipo = 0; // Indica nenhuma peça
    }//FIM INICIA_PARA
```

```
//FIM AUTO_MAN
```

```
if(HAB_DES == HIGH) analogWrite(8,170); // Auto destruicao
```

```
if(CALIBRACAO == LOW) calibracao();
```

```
if(REFERENCIAMENTO == LOW) //Executa a rotina de referenciamento
```

```
{
```

```
referencia();
```

```
delay(900);
```

```
} // Fim Referenciamento
```

```
//FIM VOID LOOP
```

```
//-----LOOP-----LOOP-----LOOP-----LOOP-----LOOP-----
```

```
//-----FUNÇÕES-----FUNÇÕES-----FUNÇÕES-----FUNÇÕES
```

```
void posicaoINI(int pos)
```

```
{
```

```
  sobe_ombro( OMBRO_P1 + 100 , 255 );
```

```
  delay(tempo);
```

```
  fecha_garra();
```

```
  delay(tempo);
```

```
  base_esquerda( 220 , pos );
```

```
  delay(tempo);
```

```
  desce_cotovelo( 150 , 50 );
```

```
  digitalWrite(51, LOW); // DESLIGA ESTEIRA DE SAIDA
```

```
//FIM posicaoINI
```

```
//=====
```

```
void posicaoP1() // PEGA PEÇA
```

```
{
```

```
desce_cotovelo( 140 , 50 );
delay(tempo);
sobe_punho( PUNHO_P1 , 195 );
delay(tempo);
leituraCOTOVELO = COTOVELO_P1 - 30; // Força a entrada no while para evitar erros de
sinais que são perdidos durante a rotina
sobe_cotovelo( COTOVELO_P1 , 155 );
delay(tempo);
abre_garra();
delay(tempo);
leituraOMBRO = OMBRO_P1 + 30; // Força a entrada no while para evitar erros de sinais
que são perdidos durante a rotina
desce_ombro( OMBRO_P1 , 60 );
delay(tempo);

} //FIM posicaoP1
//=====
void posicaoP2(int pos) // ESTEIRA
{
    sobe_ombro( (OMBRO_P1 + 60) , 255 );
    delay(tempo);
    base_direita( 210 , pos );
    delay(tempo);
    leituraCOTOVELO = COTOVELO_P1 - 30; // Força a entrada no while para evitar erros de
sinais que são perdidos durante a rotina
    sobe_cotovelo( COTOVELO_P1 + 155 , 190 );
    delay(tempo);
    leituraOMBRO = OMBRO_P1; // Força a entrada no while para evitar erros de sinais que são
perdidos durante a rotina
    desce_ombro( (OMBRO_P1 - 35) , 65 );
    digitalWrite(51, LOW); // DESLIGA ESTEIRA DE SAIDA
    delay(tempo);
    desce_punho( (PUNHO_P1 - 15) , 80 );
```

```
    delay(tempo*5);
    abre_garra();
    delay(tempo);

} //FIM posicaoP2

//=====

void posicaoP3(int pos) // CAIXA
{
    sobe_ombro( (OMBRO_P1 + 60) , 255 );
    delay(tempo);
    base_direita( 210 , pos );
    delay(tempo);
    leituraCOTOVELO = COTOVELO_P1 - 30; // Força a entrada no while para evitar erros de
    sinais que são perdidos durante a rotina
    sobe_cotovelo( COTOVELO_P1 + 100 , 190 );
    delay(tempo);
    desce_punho( (PUNHO_P1 - 200) , 70 );
    delay(tempo*5);
    leituraOMBRO = OMBRO_P1; // Força a entrada no while para evitar erros de sinais que são
    perdidos durante a rotina
    desce_ombro( (OMBRO_P1 - 15) , 70 );
    delay(tempo*3);
    abre_garra();
    delay(tempo);

} //FIM posicaoP3

//=====

void referencia()
{

//REF_CONT = REF_CONT + 1;
```

```
//
// if(REF_CONT == 1)//Posicao Inicial
// {
    protocolo[16] = 'E'; // INICIA O ENCHIMENTO DA BARRA
    posicaoINI( 1 );
// }
//
// if(REF_CONT == 2)//Posicao P1
// {
// protocolo[16] = 'E'; //Atualiza protocolo dizendo que o referenciamento está INICIANDO -
// ATUALIZA A BARRA DE CARREGAMENTO DE 0 A 10
// teste();
// }

protocolo[16] = 'O'; // FINALIZA O ENCHIMENTO DA BARRA
} //Fim referencia
//=====

void calibracao()
{
int select;
select = 0;
leituras;

if(AUTO_MAN == HIGH){ delay(500); gravar();}

while(select == 0)
{
    lcd.setCursor(0,0); lcd.print("CONTROLE  MANUAL");
    lcd.setCursor(4,1); lcd.print("CALIBRAR      ");
    lcd.setCursor(-3,2); lcd.print("          ");
    lcd.setCursor(-4,3); lcd.print("          ");
    if(REFERENCIAMENTO == LOW){ delay(500); select = select + 1;}
```

```
    leituras();
}

while(select == 1) // PUNHO SUBIR
{
    lcd.setCursor(-4,3); lcd.print("PUNHO SUBIR  ");
    if(HAB_DES == HIGH) analogWrite(2, 200 ); // Desce Ombro
    if(HAB_DES == LOW) analogWrite(2, 0 );
    if(REFERENCIAMENTO == LOW){ delay(500); select = select + 1;}
    if(AUTO_MAN == HIGH){ delay(500); gravar();}
    leituras();
}

while(select == 2) // PUNHO DESCER
{
    lcd.setCursor(-4,3); lcd.print("PUNHO DESCER  ");
    if(HAB_DES == HIGH) analogWrite(3, 200 ); // Desce Ombro
    if(HAB_DES == LOW) analogWrite(3, 0 );
    if(REFERENCIAMENTO == LOW){ delay(500); select = select + 1;}
    if(AUTO_MAN == HIGH){ delay(500); gravar();}
    leituras();
}

while(select == 3) // COTOVELO SUBIR
{
    lcd.setCursor(-4,3); lcd.print("COTOVELO SUBIR ");
    if(HAB_DES == HIGH) analogWrite(4, 200 ); // Desce Ombro
    if(HAB_DES == LOW) analogWrite(4, 0 );
    if(REFERENCIAMENTO == LOW){ delay(500); select = select + 1;}
    if(AUTO_MAN == HIGH){ delay(500); gravar();}
    leituras();
}
```

```
while(select == 4) // COTOVELO DESCER
{
  lcd.setCursor(-4,3); lcd.print("COTOVELO DESCER ");
  if(HAB_DES == HIGH) analogWrite(5, 70 ); // Desce Ombro
  if(HAB_DES == LOW) analogWrite(5, 0 );
  if(REFERENCIAMENTO == LOW){ delay(500); select = select + 1;}
  if(AUTO_MAN == HIGH){ delay(500); gravar();}
  leituras();
}
```

```
while(select == 5) // OMBRO SUBIR
{
  lcd.setCursor(-4,3); lcd.print("OMBRO SUBIR  ");
  if(HAB_DES == HIGH) analogWrite(6, 255 ); // Desce Ombro
  if(HAB_DES == LOW) analogWrite(6, 0 );
  if(REFERENCIAMENTO == LOW){ delay(500); select = select + 1;}
  if(AUTO_MAN == HIGH){ delay(500); gravar();}
  leituras();
}
```

```
while(select == 6) // OMBRO DESCER
{
  lcd.setCursor(-4,3); lcd.print("OMBRO DESCER  ");
  if(HAB_DES == HIGH) analogWrite(7, 70 ); // Desce Ombro
  if(HAB_DES == LOW) analogWrite(7, 0 );
  if(REFERENCIAMENTO == LOW){ delay(500); select = select + 1;}
  if(AUTO_MAN == HIGH){ delay(500); gravar();}
  leituras();
}
```

```
while(select == 7) // BASE DIREITA
{
  lcd.setCursor(-4,3); lcd.print("BASE DIREITA  ");
```

```

if(HAB_DES == HIGH) analogWrite(8, 210 ); // Desce Ombro
if(HAB_DES == LOW) analogWrite(8, 0 );
if(REFERENCIAMENTO == LOW){ delay(500); select = select + 1;}
if(AUTO_MAN == HIGH){ delay(500); gravar();}
leituras();
}

```

```

while(select == 8) // BASE ESQUERDA
{
  lcd.setCursor(-4,3); lcd.print("BASE ESQUERDA ");
  if(HAB_DES == HIGH) analogWrite(9, 220 ); // Desce Ombro
  if(HAB_DES == LOW) analogWrite(9, 0 );
  if(REFERENCIAMENTO == LOW){ delay(500); select = select + 1;}
  if(AUTO_MAN == HIGH){ delay(500); gravar();}
  leituras();
}

```

```

while(select == 9) // BASE ESQUERDA
{
  lcd.setCursor(-4,3); lcd.print("SAIR - APERTE B ");
  if(REFERENCIAMENTO == LOW){ delay(500); select = select + 1;}
  leituras();
}
} // Fim CALIBRACAO

```

```

//=====

```

```

void gravar()
{
  int gravar, gravado;
  gravar = 1; gravado = 0;
  lcd.clear();

```

```

while(gravar == 1) // GRAVAR POSICAO P1

```

```

{
  if(gravado == 0) lcd.setCursor(-4,3); lcd.print("POSICAO P1   ");
  if(gravado == 1) {lcd.clear(); gravado = gravado + 1;}
  if(gravado == 2) lcd.setCursor(-4,3); lcd.print("GRAVADO   ");

  if(HAB_DES == HIGH)
  {
    delay(500);
    leituras();
    OMBRO_P1 = leituraOMBRO + ganho;
    COTOVELO_P1 = leituraCOTOVELO;
    PUNHO_P1 = leituraPUNHO;
    gravado = gravado + 1;
  }

  if(REFERENCIAMENTO == LOW){ delay(500); gravar = gravar + 1; gravado = 0;}

  leituras();
}

  lcd.begin(16,4);
  lcd.setCursor(0,0); lcd.print("CONTROLE  MANUAL");
  lcd.setCursor(4,1); lcd.print("CALIBRAR   ");
  lcd.setCursor(-3,2); lcd.print("           ");
  lcd.setCursor(-4,3); lcd.print("           ");

} // Fim GRAVAR
//=====================================================

void sobe_cotovelo(int pos, int pwm)
{
  int amostra = 150;
  int vet[amostra];
  int soma, i, erro, velocidade;

```

```
analogWrite(5,0); //Desabilita descida
while(leituraCOTOVELO < pos)
{
    soma = 0; //Iniciando Filtro
    for(i = 0 ; i < amostra ; i++)
    {
        vet[i] = analogRead(A1);
        soma = soma + vet[i];
    } //Finalizando Filtro

    leituraCOTOVELO = soma/amostra;
    erro = pos - leituraCOTOVELO;
    velocidade = erro + 135;
    if(velocidade > 160) velocidade = 160;

    analogWrite(4, velocidade); //Sobe
    converte_pot(leituraCOTOVELO , 6); //Atualiza protocolo
}
analogWrite(4,60); //Fixa posicao

} //Fim sobe_cotovelo
//=====

void desce_cotovelo(int pos, int pwm)
{
    analogWrite(4,0); //Desabilita subida
    while(leituraCOTOVELO > pos)
    {
        analogWrite(5, pwm ); //Desce
        leituraCOTOVELO = analogRead(A1);
        converte_pot(leituraCOTOVELO , 6); //Atualiza protocolo
    }
}
```

```
    analogWrite(5,0);
    analogWrite(4,80); delay(50); analogWrite(4,0); //Fixa posicao

} //Fim desce_cotovelo
//=====

void sobe_punho(int pos, int pwm)
{
    int amostra = 150;
    int vet[amostra];
    int soma, i, erro, velocidade;

    analogWrite(3,0); //Desabilita descida
    while(leituraPUNHO < pos)
    {
        soma = 0;
        for(i = 0 ; i < amostra ; i++) //Iniciando Filtro
        {
            vet[i] = analogRead(A2);
            soma = soma + vet[i];
        } //Finalizando Filtro

        leituraPUNHO = soma/amostra;
        erro = pos - leituraPUNHO;
        velocidade = erro + 150;
        if(velocidade > 200) velocidade = 200;

        analogWrite(2, velocidade ); //Sobe
        converte_pot(leituraPUNHO , 9); //Atualiza protocolo
    }
    analogWrite(2,150); //Fixa posicao

} //Fim sobe_punho
```

```
//=====
```

```
void desce_punho(int pos, int pwm)
{
    analogWrite(2,0); //Desabilita subida
    while(leituraPUNHO > pos)
    {
        analogWrite(3, pwm ); //Desce
        leituraPUNHO = analogRead(A2);
        converte_pot(leituraPUNHO , 9); //Atualiza protocolo
    }
    analogWrite(3,0);
    analogWrite(2,40); //Fixa posicao

```

```
}//Fim desce_punho
```

```
//=====
```

```
void sobe_ombro(int pos, int pwm)
{
    int amostra = 200;
    int vet[amostra];
    int soma, i, velocidade, erro;
    analogWrite(7,0); //Desabilita descida
    while(leituraOMBRO < pos)
    {
        soma = 0; //Iniciando Filtro
        for(i = 0 ; i < amostra ; i++)
        {
            vet[i] = analogRead(A0);
            soma = soma + vet[i];
        } //Finalizando Filtro

        leituraOMBRO = soma/amostra;

```

```
    erro = pos - leituraOMBRO;
    velocidade = erro + 230;
    if(velocidade > 255) velocidade = 255;

    analogWrite(6, velocidade ); //Sobe

    converte_pot(leituraOMBRO , 3); //Atualiza protocolo
}
analogWrite(6,0); //Para subida

} //Fim sobe_ombro
//=====

void desce_ombro(int pos, int pwm)
{
    int amostra = 350;
    int vet[amostra];
    int soma, i;
    analogWrite(6,0); //desabilita descida
    while(leituraOMBRO > pos) //Valor da posição
    {
        analogWrite(7, pwm ); //Desce

        soma = 0; //Iniciando Filtro
        for(i = 0 ; i < amostra ; i++)
        {
            vet[i] = analogRead(A0);
            soma = soma + vet[i];
        } //Finalizando Filtro

        leituraOMBRO = soma/amostra;
        converte_pot(leituraOMBRO , 3); //Atualiza protocolo
    }
```

```
    analogWrite(7,0); //Para descida

} //Fim desce_ombro

//=====

void base_direita(int pwm, int pos)
{
    // porta 8 base direita
    int amostra = 100;
    int vet[amostra];
    int soma, i, pulsos, cont, contra;
    cont = 0;

    while(cont < pos)
    {

        if (pulsos > 700) contra = 0; else contra = 1; // começa a contar após sair da faixa preta
        analogWrite(8, pwm);
        Serial.print(" "); //Mantém a porta serial ocupada para não alterar bits de forma errada

        soma = 0; //Iniciando Filtro
        for(i = 0 ; i < amostra ; i++)
        {
            vet[i] = analogRead(A9);
            soma = soma + vet[i];
        } //Finalizando Filtro

        pulsos = soma/amostra;
        protocolo[1] = troca(1); // Andando para direita Converte para protocolo
        protocolo[2] = troca(0); // Andando para direita Converte para protocolo

        if(contra == 1)
        {
```

```
        if(pulsos > 700)
        {
            cont = cont + 1;
        }//if pulsos
    }//Fim contra

} //while

protocolo[1] = troca(0); // Braço parado  Converte para Protocolo
analogWrite(8, 0);
Serial.print(" "); //Mantém a porta serial ocupada para não alterar bits de forma errada

} //Fim base_direita

//=====

void base_esquerda(int pwm, int pos)
{
    // porta 9 base esquerda
    int amostra = 100;
    int vet[amostra];
    int vet2[amostra];
    int soma, soma2, i, ref, pulsos, cont, contra;
    cont = 1;

    while(cont > 0)
    {

        if (pulsos > 700) contra = 0; else contra = 1; // começa a contar após sair da faixa preta
        analogWrite(9, pwm);
        Serial.print(""); //Mantém a porta serial ocupada para não alterar bits de forma errada

        soma = 0; soma2 = 0; //Iniciando Filtro
```

```
indica = HIGH;
int cont = 0;
    while(cont < 30)
    {
        digitalWrite(10 , HIGH);
        delay(3);
        digitalWrite(10 , LOW);
        delay(17);
        cont = cont + 1;
    }
//GARRA.write(150);

converte_dig(indica , 12); //Atualiza protocolo
} //Fim abre_garra

//=====

void fecha_garra()
{
    indica = LOW;
    int cont = 0;
        while(cont < 40)
        {
            digitalWrite(10 , HIGH);
            delay(0);
            digitalWrite(10 , LOW);
            delay(20);
            cont = cont + 1;
        }
//GARRA.write(0);

converte_dig(indica , 12); //Atualiza protocolo
} //Fim fecha_garra

//=====
```

```
void converte_base(int cod, int casa)
{
```

```
if(cod > 800) protocolo[casa] = 'F'; // identificou recebe 1
if(cod < 800) protocolo[casa] = 'E'; // refletiu recebe 0
```

```
}//Fim converte base
```

```
//=====
```

```
void converte_dig(int sinal, int casa)
{
```

```
if(sinal == LOW) protocolo[casa] = 'E';
if(sinal == HIGH) protocolo[casa] = 'F';
```

```
}//Fim converte dig
```

```
//=====
```

```
void converte_pot(int cod, int casa)
{
```

```
int U, D, C, UNI, DEZ, CEN, i;
```

```
if(cod < 100)
{
```

```
    U = cod%10;
```

```
    D = cod/10;
```

```
    C = 0;
```

```
}//Fim if 10
```

```
if(cod >= 100)
{
```

```
    U = cod%10;
```

```
    D = (cod/10)%10;
```

```
C = (cod/10)/10;  
} //Fim if 100
```

```
UNI = troca(U);  
DEZ = troca(D);  
CEN = troca(C);
```

```
protocolo[casa] = CEN;  
protocolo[casa + 1] = DEZ;  
protocolo[casa + 2] = UNI;
```

```
} //fim converte pot
```

```
//=====
```

```
char troca(int aux)
```

```
{
```

```
char ret;
```

```
if(aux == 0) ret = 'E';  
if(aux == 1) ret = 'F';  
if(aux == 2) ret = 'G';  
if(aux == 3) ret = 'H';  
if(aux == 4) ret = 'I';  
if(aux == 5) ret = 'J';  
if(aux == 6) ret = 'K';  
if(aux == 7) ret = 'L';  
if(aux == 8) ret = 'M';  
if(aux == 9) ret = 'N';  
if(aux == 10) ret = 'O';  
if(aux == 11) ret = 'P';  
if(aux == 12) ret = 'Q';  
if(aux == 13) ret = 'R';
```

```
if(aux == 14) ret = 'S';
if(aux == 15) ret = 'T';
if(aux == 16) ret = 'U';
if(aux == 17) ret = 'V';
if(aux == 18) ret = 'W';
if(aux == 19) ret = 'X';
if(aux == 20) ret = 'Y';
if(aux == 21) ret = 'Z';

return(ret);
} //Fim troca

//=====

void teste() //TESTA OS TRANSISTORES PARA ESQUENTÁ-LOS
{
int cont, TEMPO;
cont = 0; TEMPO = 500;

sobe_cotovelo(180 , 155); delay(800); //INICIO DO TESTE
sobe_punho(690 , 195);

HAB_DES = digitalRead(33);
if(HAB_DES == HIGH) cont = 3; //Desabilitar

while(cont < 3) // TESTANDO OMBRO
{
if (cont == 0) protocolo[16] = 'F'; if (cont == 1) protocolo[16] = 'G'; if (cont == 2)
protocolo[16] = 'H'; // Atualiza protocolo para preenchimento da barra de 0 a 10
desce_ombro(OMBRO_P1 , 70);
delay(TEMPO*2);
sobe_ombro(OMBRO_P1 + 100 , 255);
delay(TEMPO);
```

```
    cont = cont +1;
    if (cont == 0) protocolo[16] = 'F'; if (cont == 1) protocolo[16] = 'G'; if (cont == 2)
    protocolo[16] = 'H'; // Atualiza protocolo para preenchimento da barra de 0 a 10
    }

cont = 0;  HAB_DES = digitalRead(33); if(HAB_DES == HIGH)cont = 3; //Desabilitar

while(cont < 3) // TESTANDO COTOVELO
{
    if (cont == 0) protocolo[16] = 'T'; if (cont == 1) protocolo[16] = 'J'; if (cont == 2)
    protocolo[16] = 'K';// Atualiza protocolo para preenchimento da barra de 0 a 10
    sobe_cotovelo(COTOVELO_P1 + 100 , 170);
    delay(TEMPO);
    desce_cotovelo(COTOVELO_P1 - 100, 50);
    delay(TEMPO);
    cont = cont +1;
    if (cont == 0) protocolo[16] = 'T'; if (cont == 1) protocolo[16] = 'J'; if (cont == 2)
    protocolo[16] = 'K';// Atualiza protocolo para preenchimento da barra de 0 a 10
    }

cont = 0;  HAB_DES = digitalRead(33); if(HAB_DES == HIGH)cont = 3; //Desabilitar

while(cont < 3) // TESTANDO PUNHO
{
    desce_punho(PUNHO_P1 - 250 , 115);
    delay(TEMPO);
    sobe_punho(PUNHO_P1 + 15, 205);
    delay(TEMPO);
    cont = cont +1;
    if (cont == 0) protocolo[16] = 'L'; if (cont == 1) protocolo[16] = 'M'; if (cont == 2)
    protocolo[16] = 'N'; // Atualiza protocolo para preenchimento da barra de 0 a 10
    }


```

```
//Fim testes
//=====================================================

void enviar_dados()
{
  cont_interrupt = cont_interrupt + 1; //A interrupção é acionada por uma frequencia de 1KHz -
  1000Hz

  if(cont_interrupt == 100) //Define o tempo de interrupcao
  {
    for(i = 0 ; i < 22 ; i++) //Percorre as 21 posições do vetor "PROTOCOLO"
      Serial.write(protocolo[i]);

    Serial.println("");
    conta_interrupcao = conta_interrupcao + 1; // Contagem para dar 1s e atualizar o lcd
    cont_interrupt = 0;
  }

  if(conta_interrupcao == 50)
  {
    leituras();
    lcd_cont = lcd_cont + 1;
    if(CALIBRACAO == HIGH) atualiza_lcd();
    conta_interrupcao = 0;
  }

} // Fim da interrupção
//=====================================================

void atualiza_lcd()
{
  if(lcd_cont > 4) lcd_cont = 1;
```

```
if(lcd_cont == 1)
{
  lcd.clear();
  lcd.begin(16, 4);
  lcd.setCursor(2,0); lcd.print("ENG.ELETRICA");
  lcd.setCursor(1,1); lcd.print("IFG ITUMBIARA");
  lcd.setCursor(-3,2); lcd.print("BRACO ROBOTICO");
  lcd.setCursor(-4,3); lcd.print("----ANO 2015----");
}
```

```
if(lcd_cont == 2)
{
  lcd.clear();
  lcd.begin(16, 4);
  lcd.setCursor(2,0); lcd.print("RESPONSAVEL");
  lcd.setCursor(0,1); lcd.print("-----");
  lcd.setCursor(-3,2); lcd.print("** FABRICIO **");
  lcd.setCursor(-3,3); lcd.print("** MENDONCA **");
}
```

```
if(lcd_cont == 3)
{
  lcd.clear();
  lcd.begin(16, 4);
  lcd.setCursor(0,0); lcd.print("PROF. ORIENTADOR");
  lcd.setCursor(0,1); lcd.print("---      ---");
  lcd.setCursor(0,2); lcd.print("GHUNTER");
  lcd.setCursor(1,3); lcd.print("PAULO");
}
```

```
if(lcd_cont == 4)
{
```

```
lcd.clear();
lcd.begin(16, 4);
lcd.setCursor(1,0); lcd.print("AGRADECIMENTOS");
lcd.setCursor(0,1); lcd.print("#          #");
lcd.setCursor(-4,2); lcd.print("##  RICARDO  ##");
lcd.setCursor(-4,3); lcd.print("###  TIRONE  ###");
lcd_cont = 0;
}
```

```
// Fim Atualização do LCD
```

```
//=====
```

```
void desabilita_portas()
```

```
{
```

```
digitalWrite(49, LOW); // Esteira de alimentação
```

```
converte_dig(LOW , 19); // Converte para colocar no vetor de protocolo
```

```
digitalWrite(51, LOW); // Esteira Saída 1
```

```
digitalWrite(53, LOW); // Esteira Saida 2
```

```
analogWrite(2,0);
```

```
analogWrite(3,0);
```

```
analogWrite(4,0);
```

```
analogWrite(5,0);
```

```
analogWrite(6,0);
```

```
analogWrite(7,0);
```

```
analogWrite(8,0);
```

```
analogWrite(9,0);
```

```
//Fim desbilita_portas
```

```
//=====
```

```
void leituras()
```

```
{
```

```
protocolo[0] = 'A'; protocolo[21] = 'D'; // Inicio e fim do protocolo de comunicacao
```

```
// ----- Digitais -----
```

```
AUTO_MAN = digitalRead(25);
```

```
converte_dig(AUTO_MAN , 13); // Converte para colocar no vetor de protocolo
```

```
INICIA_PARA = digitalRead(23);
```

```
converte_dig(INICIA_PARA , 14); // Converte para colocar no vetor de protocolo
```

```
HAB_DES = digitalRead(33);
```

```
converte_dig(HAB_DES , 15); // Converte para colocar no vetor de protocolo
```

```
REFERENCIAMENTO = digitalRead(31);
```

```
protocolo[16] = troca(10); // BARRA CARREGADA 100%
```

```
CALIBRACAO = digitalRead(52);
```

```
converte_dig(indica , 12); // Converte GARRA para colocar no vetor de protocolo
```

```
protocolo[20] = troca(tipo); // Converte ERRO para colocar no vetor de protocolo
```

```
SENSOR_CAP = digitalRead(29);
```

```
protocolo[17] = troca(PLASTICOS); // Insere no protocolo
```

```
SENSOR_IND = digitalRead(27);
```

```
protocolo[18] = troca(METAIS); // Insere no protocolo
```

```
//converte_dig(METAIS , 18); // Converte para colocar no vetor de protocolo
```

```
// ----- Analogicas -----
```

```
leituraOMBRO = analogRead(A0);
```

```
converte_pot(leituraOMBRO , 3); // Converte para colocar no vetor de protocolo
```

```
leituraCOTOVELO = analogRead(A1);
converte_pot(leituraCOTOVELO , 6); // Converte para colocar no vetor de protocolo

leituraPUNHO = analogRead(A2);
converte_pot(leituraPUNHO , 9); // Converte para colocar no vetor de protocolo

pulsos = analogRead(A9);
protocolo[1] = troca(0); // Braço parado  Converte para Protocolo
//
ref = analogRead(A10);
protocolo[2] = troca(0); // Braço parado  Converte para Protocolo

} //Fim leituras

//-----FUNÇÕES-----FUNÇÕES-----FUNÇÕES-----FUNÇÕES
```

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Pereira, S. B. **A nova Indústria, a Robótica, e as mudanças na Economia**. 2008. Disponível na Internet: <<https://www.linkedin.com/pulse/nova-ind%C3%BAstria-rob%C3%B3tica-e-mudan%C3%A7as-na-economia-sudanes-b-pereira>>. Acesso em: 15 nov. 2015.
- [2] Fitzgerald, A. E; KINGSLEY, Jr. C; Umans, S. D. **Máquinas Elétricas**. 6. ed: Bookman, 2006. 177 p. 369p. 177p. 178p. 179 p. 295p.
- [3] Kosow, I. L. **Máquinas Elétricas e Transformadores**. 3. ed: Globo, 1979. 39 p. 40p.
- [4] Mendonça, V. **SERVOMOTOR: Princípios básicos de funcionamento**. Disponível na Internet: <<http://www.ebah.com.br/content/ABAAAAaLkAB/servomotor-principios-basicos-funcionamento#>>. Acesso em: 28 nov. 2015.
- [5] Carvalho, L. **O que são servomotores e como eles funcionam?**. Disponível na Internet: <<http://electricconcept.blogspot.com.br/2015/04/o-que-sao-servomotores-e-como-funcionam.html>>. Acesso em: 28 nov. 2015.
- [6] Rosário, J. M. **Princípios de Mecatrônica**. 1. ed. São Paulo: Pearson, 2005. 4 cap. 2 cap. 6 cap. 8 cap.
- [7] Cristiam. **Características de sensores ópticos**. Disponível na Internet: <<http://sensoresopticos.blogspot.com.br/2012/03/los-sensores-se-les-debe-exigir-la.html>>. Acesso em: 28 nov. 2015.
- [8] **Sensores de proximidade magnéticos**. Disponível na Internet: <http://www.ffonseca.com/produto.aspx?lang=pt&id_object=673521&name=Sensores-de-proximidade-magneticos---Sick>. Acesso em: 28 nov. 2015.
-

-
- [9] **Microprocessadores**. Disponível na Internet: <<http://iris.sel.eesc.usp.br/sel433a/Micros.pdf>>. Acesso em: 01 nov. 2015.
- [10] **Projetos com microcontrolador PIC**. Disponível na Internet: <<http://www.dm.ufscar.br/profs/waldeck/pic/>>. Acesso em: 06 nov. 2015.
- [11] **Guetting Started with the Arduino DUE**. Disponível na Internet: <<http://arduino.cc/en/Guide/ArduinoDue>>. Acesso em: 06 ago. 2014.
- [12] Mecatrônica Atual. **SSC – Sistemas de Supervisão e Controle**. Disponível na Internet: <<http://www.mecatronicaatual.com.br/educacao/1124-ssc-sistemas-de-superviso-e-controle%20>>. Acesso em: 07 nov. 2015.
- [13] **Supervisório SCADA – Software para aquisição de dados**. Disponível na Internet: <<https://www.youtube.com/watch?v=oPa6fw0if1c>>. Acesso em: 28 nov. 2015.
- [14] National Instruments. **Ambiente gráfico de desenvolvimento de sistemas LabVIEW**. Disponível na Internet: <<http://www.ni.com/labview/pt/>>. Acesso em: 07 nov. 2015.
- [15] Universidade Tecnológica Federal do Paraná. Núcleo de Pesquisas em Eletrotécnica. **Introdução a Programação com LabVIEW™**. Curitiba: 2014.
- [16] Leandro, A. **Tutorial de configuração padrão para módulo transmissores e receptores protocolo zigbee e comunicação PIC16f877A**. Disponível na internet: <<https://engleandroalves.wordpress.com/2012//>>. Acesso em: 29 nov. 2015.
- [17] Nerys, J. W. L. **Microprocessadores e Microcontroladores**. Universidade Federal de Goiás: Núcleo de Estudo e Pesquisa em Processamento de Energia e Qualidade.
- [18] Hasse, D. **Microcontroladores e Microprocessadores, Apostila 01**. Universidade Paulista: São José dos Campos.
-

- [19] PEDROSO, L. J. **Projeto de Placas de Circuito Impresso**. Escola Técnica Federal do Mato Grosso.
- [20] National Instruments. **Robotcs Simulations Utility Using 3D Picture Control**. 2008. Disponível na Internet: < <https://decibel.ni.com/content/docs/DOC-2430>>. Acesso em: 15 nov. 2015.
- [21] CRAIG. J. J. **Introduction to Robotics Mechanics and Control**. 3. ed. New Jersey: Pearson, 2005.
- [22] PATSKO, L. F. **Tutorial Montagem da Ponte H**. Maxwell Bhor, Instrumentação Eletrônica, 2006.
- [23] Mabuchi Motor: Automotive Appliances. **GD-558RC/LC**. Disponível na Internet: <http://www.mabuchi-motor.co.jp/cgi-in/catalog/e_catalog.cgi?CAT_ID=gd_558rclc>. Acesso em: 07 nov. 2015.
- [24] Semiconductor Technical Data. **Darlington Complementary Silicon Power Transistors – TIP142/147**. Motorola, 1995.
- [25] Mecatrônica Atual. **Servo-acionamentos**. Disponível na Internet: < <http://www.mecatronicaatual.com.br/educacao/1038-servo-acionamentos> >. Acesso em: 18 ago. 2015.
- [26] **Transistores**. Disponível na Internet: <<http://www.cear.ufpb.br/~mauricio/wp-content/uploads/2015/05/Transistor.pdf>>. Acesso em: 05 nov. 2015.
- [27] **Paletização Robotizada**. Disponível na Internet: <<http://www.torfresma.com.br/produtos/areas-de-atuacao/automacao-e-robotica/linhas/paletizacao-robotizada>>. Acesso em: 06 nov. 2015.
- [28] SANTOS, P. **Introdução ao Arduino**. ASPOF EM-AEL.
-